Coded Cooperative Data Exchange in Multihop Networks

Thomas A. Courtade, *Student Member, IEEE,*
and Richard D. Wesel, *Senior Member, IEEE*

arXiv:1203.3445v1 [cs.IT] 15 Mar 2012

Abstract

Consider a connected network of n nodes that all wish to recover k desired packets. Each node begins with a subset of the desired packets and exchanges coded packets with its neighbors. This paper provides necessary and sufficient conditions which characterize the set of all transmission schemes that permit every node to ultimately learn (recover) all k packets. When the network satisfies certain regularity conditions and packets are randomly distributed, this paper provides tight concentration results on the number of transmissions required to achieve universal recovery. For the case of a fully connected network, a polynomial-time algorithm for computing an optimal transmission scheme is derived. An application to secrecy generation is discussed.

Index Terms

Coded Cooperative Data Exchange, Universal Recovery, Network Coding.

I. Introduction

Consider a connected network of n nodes that all wish to recover k desired packets. Each node begins with a subset of the desired packets and broadcasts messages to its neighbors over discrete, memoryless, and interference-free channels. Furthermore, every node knows which packets are already known by each node and knows the topology of the network. How many

The authors are with the Electrical Engineering Department, University of California, Los Angeles, CA, 90095 USA (email: tacourta@ee.ucla.edu; wesel@ee.ucla.edu).

This work was presented in part at the 2010 Military Communications Conference (MILCOM) [1] and the 2010-2011 Allerton Conference on Communication, Control, and Computing [2], [3].

This research was supported by Rockwell Collins through contract #4502769987.

transmissions are required to disseminate the k packets to every node in the network? How should this be accomplished? These are the essential questions addressed. We refer to this as the *Coded Cooperative Data Exchange* problem, or just the Cooperative Data Exchange problem.

This work is motivated in part by emerging issues in distributed data storage. Consider the problem of backing up data on servers in a large data center. One commonly employed method to protect data from corruption is replication. Using this method, large quantities of data are replicated in several locations so as to protect from various sources of corruption (e.g., equipment failure, power outages, natural disasters, etc.). As the quantity of information in large data centers continues to increase, the number of file transfers required to complete a periodic replication task is becoming an increasingly important consideration due to time, equipment, cost, and energy constraints. The results contained in this paper address these issues.

This model also has natural applications in the context of tactical networks, and we give one of them here. Consider a scenario in which an aircraft flies over a group of nodes on the ground and tries to deliver a video stream. Each ground node might only receive a subset of the transmitted packets due to interference, obstructions, and other signal integrity issues. In order to recover the transmission, the nodes are free to communicate with their neighbors, but would like to minimize the number of transmissions in order to conserve battery power (or avoid detection, etc.). How should the nodes share information, and what is the minimum number of transmissions required so that the entire network can recover the video stream?

Beyond the examples mentioned above, the results presented herein can also be applied to practical secrecy generation amongst a collection of nodes. We consider this application in detail in Section IV.

A. Related Work

Distributed data exchange problems have received a great deal of attention over the past several years. The powerful techniques afforded by network coding [4], [5] have paved the way for cooperative communications at the packet-level.

The coded cooperative data exchange problem (also called the universal recovery problem in [1]–[3]) was originally introduced by El Rouayheb *et al.* in [6], [7] for a fully connected network (i.e., a single-hop network). For this special case, a randomized algorithm for finding an optimal transmission scheme was given in [8], and the first deterministic algorithm was recently given in [9]. In the concluding remarks of [9], the weighted universal recovery problem (in which the objective is to minimize the weighted sum of transmissions by nodes) was posed as an open problem. However, this was solved using a variant of the same algorithm in [10], and independently by the present authors using a submodular algorithm in [3].

The coded cooperative data exchange problem is related to the index coding problem originally introduced by Birk and Kol in [11]. Specifically, generalizing the index coding problem to permit each node to be a transmitter (instead of having a single server) and further generalizing so that the network need not be a single hop network leads to a class of problems that includes our problem as a special case in which each node desires to receive all packets.

One significant result in index coding is that nonlinear index coding outperforms the best linear index code in certain cases [12], [13]. As discussed above, our problem is a special case of the generalized index coding problem, and it turns out that linear encoding does achieve the minimum number of transmissions required for universal recovery and this solution is computable in polynomial time for some important cases.

This paper applies principles of cooperative data exchange to generate secrecy in the presence of an eavesdropper. In this context, the secrecy generation problem was originally studied in [14]. In [14], Csiszar and Narayan gave single-letter characterizations of the secret-key and private-key capacities for a network of nodes connected by an error-free broadcast channel. While general and powerful, these results left two practical issues as open questions. First, (as with many information-theoretic investigations) the results require the nodes to observe arbitrarily long sequences of i.i.d. source symbols, which is generally not practical. Second, no efficient algorithm is provided in [14] which achieves the respective secrecy capacities. More recent work in [15], [16] addressed the latter point.

B. Our Contributions

In this paper, we provide necessary and sufficient conditions for achieving universal recovery[1] in arbitrarily connected multihop networks. We specialize these necessary and sufficient conditions to obtain precise results in the case where the underlying network topology satisfies some modest regularity conditions.

[1] In this paper, we use the term *universal recovery* to refer to the ultimate condition where every node has successfully recovered all packets.

For the case of a fully connected network, we provide an algorithm based on submodular optimization which solves the cooperative data exchange problem. This algorithm is unique from the others previously appearing in the literature (cf. [8]–[10]) in that it exploits submodularity. As a corollary, we provide exact concentration results when packets are randomly distributed in a network.

In this same vein, we also obtain tight concentration results and approximate solutions when the underlying network is d-regular and packets are distributed randomly.

Furthermore, if packets are divisible (allowing transmissions to consist of partial packets), we prove that the traditional cut-set bounds can be achieved for any network topology. In the case of d-regular and fully connected networks, we show that splitting packets does not typically provide any significant benefits.

Finally, for the application to secrecy generation, we leverage the results of [14] in the context of the cooperative data exchange problem for a fully connected network. In doing so, we provide an efficient algorithm that achieves the secrecy capacity without requiring any quantities to grow asymptotically large.

C. Organization

This paper is organized as follows. Section II formally introduces the problem and provides basic definitions and notation. Section III presents our main results. Section IV discusses the application of our results to secrecy generation by a collection of nodes in the presence of an eavesdropper. Section V contains the relevant proofs. Section VI delivers the conclusions and discusses directions for future work.

II. System Model and Definitions

Before we formally introduce the problem, we establish some notation. Let $\mathbb{N} = 0, 1, 2, \dots$ denote the set of natural numbers. For two sets A and B, the relation $A \subset B$ implies that A is a proper subset of B (i.e., $A \subseteq B$ and $A \neq B$). For a set A, the corresponding power set is denoted $2^A := \{B : B \subseteq A\}$. We use the notation $[m]$ to denote the set $\{1, \dots, m\}$.

This paper considers a network of n nodes. The network must be connected, but it need not be fully connected (i.e., it need not be a complete graph). A graph $\mathcal{G} = (V, E)$ describes the specific connections in the network, where V is the set of vertices $\{v_i : i \in \{1, \dots, n\}\}$ (each corresponding to a node) and E is the set of edges connecting nodes. We assume that the edges in E are undirected, but our results can be extended to directed graphs.

Each node wishes to recover the same k desired packets, and each node begins with a (possibly empty) subset of the desired packets. Formally, let $P_i \subseteq \{p_1, \dots, p_k\}$ be the (indexed) set of packets originally available at node i, and $\{P_i\}_{i=1}^n$ satisfies $\bigcup_{i=1}^n P_i = \{p_1, \dots, p_k\}$. Each $p_j \in \mathbb{F}$, where $\mathbb{F}$ is some finite field (e.g. $\mathbb{F} = \mathrm{GF}(2^m)$). For our purposes, it suffices to assume $|\mathbb{F}| \geq 2n$. The set of packets initially missing at node i is denoted $P_i^c := \{p_1, \dots, p_k\} \backslash P_i$.

Throughout this paper, we assume that each packet $p_i \in \{p_1, \dots, p_k\}$ is equally likely to be any element of $\mathbb{F}$. Moreover, we assume that packets are independent of one another. Thus, no correlation between different packets or prior knowledge about unknown packets can be exploited.

To simplify notation, we will refer to a given problem instance (i.e., a graph and corresponding sets of packets available at each node) as a network $\mathcal{T} = \{\mathcal{G}, P_1, \dots, P_n\}$. When no ambiguity is present, we will refer to a network by $\mathcal{T}$ and omit the implicit dependence on the parameters $\{\mathcal{G}, P_1, \dots, P_n\}$.

Let the set $\Gamma(i)$ be the neighborhood of node i. There exists an edge $e \in E$ connecting two vertices $v_i, v_j \in V$ iff $i \in \Gamma(j)$. For convenience, we put $i \in \Gamma(i)$. Node i sends (possibly coded) packets to its neighbors $\Gamma(i)$ over discrete, memoryless, and interference-free channels. In other words, if node i transmits a message, then every node in $\Gamma(i)$ receives that message. If S is a set of nodes, then we define $\Gamma(S) = \cup_{i \in S} \Gamma(i)$. In a similar manner, we define $\partial(S) = \Gamma(S) \backslash S$ to be the boundary of the vertices in S. An example of sets S, $\Gamma(S)$, and $\partial(S)$ is given in Figure 1.

This paper seeks to determine the minimum number of transmissions required to achieve universal recovery (when every node has learned all k packets). We primarily consider the case where packets are deemed indivisible. In this case, a single transmission by user i consists of sending a packet (some $z \in \mathbb{F}$) to all nodes $j \in \Gamma(i)$. This motivates the following definition.

Definition 1: Given a network $\mathcal{T}$, the minimum number of transmissions required to achieve universal recovery is denoted $M^*(\mathcal{T})$.

To clarify this concept, we briefly consider two examples:

Example 1 (Line Network): Suppose $\mathcal{T}$ is a network of nodes connected along a line as follows: $V = \{v_1, v_2, v_3\}$, $E = \{(v_1, v_2), (v_2, v_3)\}$, $P_1 = \{p_1\}$, $P_2 = \emptyset$, and $P_3 = \{p_2\}$. Note

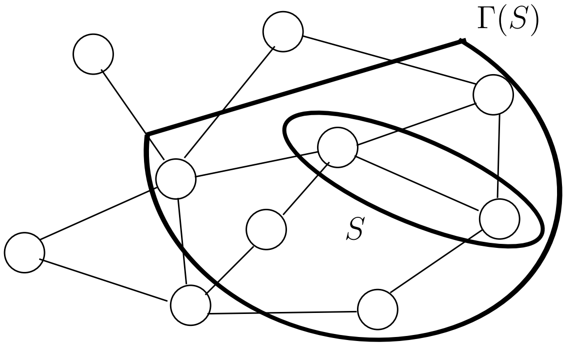

Fig. 1. For the given graph, a set of vertices S and its neighborhood $\Gamma(S)$ are depicted. The set $\partial(S)$ (i.e., the boundary of S) consists of the four vertices in $\Gamma(S)$ which are not in S.

that each node must transmit at least once in order for all nodes to recover $\{p_1, p_2\}$, hence $M^*(\mathcal{T}) \geq 3$. Suppose node 1 transmits p_1 and node 3 transmits p_2. Then (upon receipt of p_1 and p_2 from nodes 1 and 3, respectively) node 2 transmits $p_1 \oplus p_2$, where $\oplus$ indicates addition in the finite field $\mathbb{F}$. This strategy requires 3 transmissions and allows each user to recover $\{p_1, p_2\}$. Hence $M^*(\mathcal{T}) = 3$.

Example 1 demonstrates a transmission schedule that uses two *rounds* of communication. The transmissions by node i in a particular round of communication can depend only on the information available to node i prior to that round (i.e. P_i and previously received transmissions from neighboring nodes). In other words, the transmissions are causal. The transmission scheme employed in Example 1 is illustrated in Figure 2.

Example 2 (Fully Connected Network): Suppose $\mathcal{T}$ is a 3-node fully connected network in which $\mathcal{G}$ is a complete graph on 3 vertices, and $P_i = \{p_1, p_2, p_3\} \backslash p_i$. Clearly one transmission is not sufficient, thus $M^*(\mathcal{T}) \geq 2$. It can be seen that two transmissions suffice: let node 1 transmit p_2 which lets node 2 have $P_2 \cup p_2 = \{p_1, p_2, p_3\}$. Now, node 2 transmits $p_1 \oplus p_3$, allowing nodes 1 and 3 to each recover all three packets. Thus $M^*(\mathcal{T}) = 2$. Since each transmission was only a function of the packets originally available at the corresponding node, this transmission strategy can be accomplished in a single round of communication.

In the above examples, we notice that the transmission schemes are partially characterized by a schedule of which nodes transmit during which round of communication. We formalize this notion with the following definition:

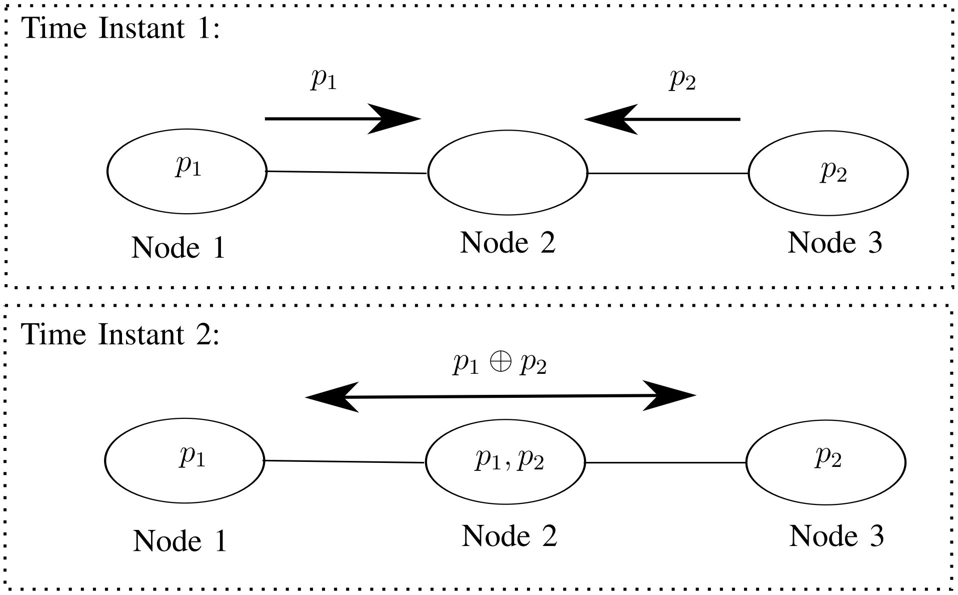

Fig. 2. An illustration of the transmission scheme employed in Example 1. During the first time instant, Nodes 1 and 3 broadcast packets p_1 and p_2, respectively. During the second time instant, Node 2 broadcasts the XOR of packets p_1 and p_2. This scheme requires three transmissions and achieves universal recovery.

Definition 2 (Transmission Schedule): A set of integers $\{b_i^j : i \in [n], j \in [r], b_i^j \in \mathbb{N}\}$ is called a transmission schedule for r rounds of communication if node i makes exactly b_i^j transmissions during communication round j.

When the parameters n and r are clear from context, a transmission schedule will be denoted by the shorthand notation $\{b_i^j\}$. Although finding a transmission schedule that achieves universal recovery is relatively easy (e.g., each node transmits all packets in their possession at each time instant), finding one that achieves universal recovery with $M^*(\mathcal{T})$ transmissions can be extremely difficult. This is demonstrated by the following example:

Example 3 (Optimal Cooperative Data Exchange is NP-Hard.): Suppose $\mathcal{T}$ is a network with $k = 1$ corresponding to a bipartite graph with left and right vertex sets V_L and V_R respectively. Let $P_i = p_1$ for each $i \in V_L$, and let $P_i = \emptyset$ for each $i \in V_R$. In this case, $M^*(\mathcal{T})$ is given by the minimum number of sets in $\{\Gamma(i)\}_{i \in V_L}$ which cover all vertices in V_R. Thus, finding $M^*(\mathcal{T})$ is at least as hard as the Minimum Set Cover problem, which is NP-complete [17].

Several of our results are stated in the context of *randomly distributed packets*. Assume $0 < q < 1$ is given. Our model is essentially that each packet is available independently at each node

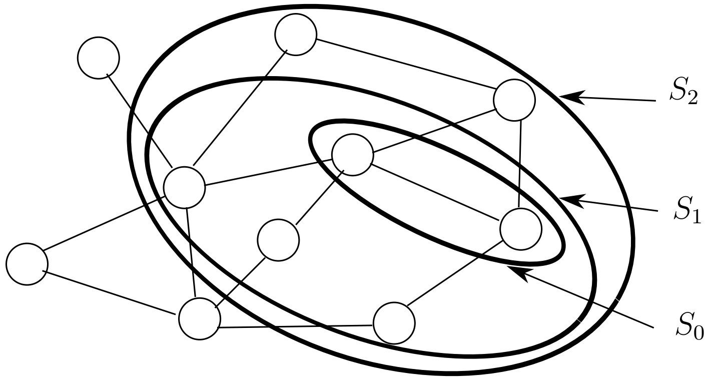

Fig. 3. An example of a sequence $(S_0, S_1, S_2) \in \mathcal{S}^{(2)}(\mathcal{G})$ for a particular choice of graph $\mathcal{G}$.

with probability q. However, we must condition on the event that each packet is available to at least one node. Thus, when packets are randomly distributed, the underlying probability measure is given by

$$\Pr\left[p_i \in \bigcup_{j\in S} P_j\right] = \frac{1-(1-q)^{|S|}}{1-(1-q)^n} \tag{1}$$

for all $i \in [k]$ and all nonempty $S \subseteq V = [n]$.

Finally, we introduce one more definition which links the network topology with the number of communication rounds, r.

Definition 3: For a graph $\mathcal{G} = (V, E)$ on n vertices, define $\mathcal{S}^{(r)}(\mathcal{G}) \subset (2^V)^{r+1}$ as follows: $(S_0, S_1, \dots, S_r) \in \mathcal{S}^{(r)}(\mathcal{G})$ if and only if the sets $\{S_i\}_{i=0}^r$ satisfy the following two conditions:

$$\emptyset \subset S_i \subset V \quad \text{for each } 0 \le i \le r \text{, and}$$

$$S_{i-1} \subseteq S_i \subseteq \Gamma(S_{i-1}) \quad \text{for each } 1 \le i \le r.$$

In words, any element in $\mathcal{S}^{(r)}(\mathcal{G})$ is a nested sequence of subsets of vertices of $\mathcal{G}$. Moreover, the constraint that each set in the sequence is contained in its predecessor's neighborhood implies that the sets cannot expand too quickly relative to the topology of $\mathcal{G}$.

To make the definition of $\mathcal{S}^{(r)}(\mathcal{G})$ more concrete, we have illustrated a sequence $(S_0, S_1, S_2) \in \mathcal{S}^{(2)}(\mathcal{G})$ for a particular choice of graph $\mathcal{G}$ in Figure 3.

III. Main Results

In this section, we present our main results. Proofs are delayed until Section V.

A. Necessary and Sufficient Conditions for Universal Recovery

First, we provide necessary and sufficient conditions for achieving universal recovery in a network $\mathcal{T}$. It turns out that these conditions are characterized by a particular set of transmission schedules $\mathcal{R}_r(\mathcal{T})$ which we define as follows:

Definition 4: For a network $\mathcal{T} = \{\mathcal{G}, P_1, \dots, P_n\}$, define the region $\mathcal{R}_r(\mathcal{T}) \subseteq \mathbb{N}^{n \times r}$ to be the set of all transmission schedules $\{b_i^j\}$ satisfying:

$$\sum_{j=1}^{r} \sum_{i \in S_j^c \cap \Gamma(S_{j-1})} b_i^{(r+1-j)} \geq \left| \bigcap_{i \in S_r} P_i^c \right| \quad \text{for each } (S_0, \dots, S_r) \in \mathcal{S}^{(r)}(\mathcal{G}).$$

Theorem 1: For a network $\mathcal{T}$, a transmission schedule $\{b_i^j\}$ permits universal recovery in r rounds of communication if and only if $\{b_i^j\} \in \mathcal{R}_r(\mathcal{T})$.

Theorem 1 reveals that the set of transmission schedules permitting universal recovery is characterized precisely by the region $\mathcal{R}_r(\mathcal{T})$. In fact, given a transmission schedule in $\mathcal{R}_r(\mathcal{T})$, a corresponding coding scheme that achieves universal recovery can be computed in polynomial time using the algorithm in [18] applied to the network coding graph discussed in the proof of Theorem 1. Alternatively, one could employ random linear network coding over a sufficiently large field size [19]. If transmissions are made in a manner consistent with a schedule in $\mathcal{R}_r(\mathcal{T})$, universal recovery will be achieved with high probability.

Thus, the problem of achieving universal recovery with the minimum number of transmissions reduces to solving a combinatorial optimization problem over $\mathcal{R}_r(\mathcal{T})$. As this problem was shown to be NP-hard in Example 3, we do not attempt to solve it in its most general form. Instead, we apply Theorem 1 to obtain surprisingly simple characterizations for several cases of interest.

Before proceeding, we provide a quick example showing how the traditional cut-set bounds can be recovered from Theorem 1.

Example 4 (Cut-Set Bounds): Considering the constraint defining $\mathcal{R}_r(\mathcal{T})$ in which the nested subsets that form $\mathcal{S}^{(r)}(\mathcal{G})$ are all identical. That is, $(S, S, \dots, S) \in \mathcal{S}^{(r)}(\mathcal{G})$ for some nonempty $S \subset V$. We see that any transmission schedule $\{b_i^j\} \in \mathcal{R}_r(\mathcal{T})$ must satisfy the familiar cut-set bounds:

$$\sum_{j=1}^{r} \sum_{i \in \partial(S)} b_i^j \geq \left| \bigcap_{i \in S} P_i^c \right|. \tag{2}$$

In words, the total number of packets that flow into the set of nodes S must be greater than or equal to the number of packets that the nodes in S are collectively missing.

B. Fully Connected Networks

When $\mathcal{T}$ is a fully connected network, the graph $\mathcal{G}$ is a complete graph on n vertices. This is perhaps one of the most practically important cases to consider. For example, in a wired computer network, clients can multicast their messages to all other terminals which are cooperatively exchanging data. In wireless networks, broadcast is a natural transmission mode. Indeed, there are protocols tailored specifically to wireless networks which support reliable network-wide broadcast capabilities (cf. [20]–[23]). It is fortunate then, that the cooperative data exchange problem can be solved in polynomial time for fully connected networks:

Theorem 2: For a fully connected network $\mathcal{T}$, a transmission schedule requiring only $M^*(\mathcal{T})$ transmissions can be computed in polynomial time. Necessary and sufficient conditions for universal recovery in this case are given by the cut-set constraints (2). Moreover, a single round of communication is sufficient to achieve universal recovery with $M^*(\mathcal{T})$ transmissions.

For the fully connected network in Example 2, we remarked that only one round of transmission was required. Theorem 2 states that this trend extends to any fully connected network.

An algorithm for solving the cooperative data exchange problem for fully connected networks is presented in Appendix A. We remark that the algorithm is sufficiently general that it can also solve the cooperative data exchange problem where the objective is to minimize the weighted sum of nodes' transmissions.

Although Theorem 2 applies to arbitrary sets of packets $P_1, \dots, P_n$, it is insightful to consider the case where packets are randomly distributed in the network. In this case, the minimum number of transmissions required for universal recovery converges in probability to a simple function of the (random) sets $P_1, \dots, P_n$.

Theorem 3: If $\mathcal{T}$ is a fully connected network and packets are randomly distributed, then

$$M^*(\mathcal{T}) = \left\lceil \frac{1}{n-1} \sum_{i=1}^{n} |P_i^c| \right\rceil .$$

with probability approaching 1 as the number of packets $k \to \infty$.

C. d-Regular Networks

Given that precise results can be obtained for fully connected networks, it is natural to ask whether these results can be extended to a larger class of networks which includes fully connected networks as a special case. In this section, we partially answer this question in the affirmative. To this end, we define d-regular networks.

Definition 5 (d-Regular Networks): A network $\mathcal{T}$ is said to be d-regular if $\partial(i) = d$ for each $i \in V$ and $\partial(S) \geq d$ for each nonempty $S \subset V$ with $|S| \leq n - d$. In other words, a network $\mathcal{T}$ is d-regular if the associated graph $\mathcal{G}$ is d-regular and d-vertex-connected.

Immediately, we see that the class of d-regular networks includes fully connected networks as a special case with $d = n - 1$. Further, the class of d-regular networks includes many frequently studied network topologies (e.g., cycles, grids on tori, etc.).

Unfortunately, the deterministic algorithm of Theorem 2 does not appear to extend to d-regular networks. However, a slightly weaker concentration result similar to Theorem 3 can be obtained when packets are randomly distributed. Before stating this result, consider the following Linear Program (LP) with variable vector $x \in \mathbb{R}^n$ defined for a network $\mathcal{T}$:

$$\text{minimize} \quad \sum_{i=1}^{n} x_i \tag{3}$$

$$\text{subject to:} \quad \sum_{i \in \partial(j)} x_i \geq \left|P_j^c\right| \quad \text{for each } j \in V. \tag{4}$$

Let $M_{LP}(\mathcal{T})$ denote the optimal value of this LP. Interpreting x_i as $\sum_j b_i^j$, the constraints in the LP are a subset of the cut-set constraints of (2) which are a subset of the necessary constraints for universal recovery given in Theorem 1. Furthermore, the integer constraints on the x_i's are relaxed. Thus $M_{LP}(\mathcal{T})$ certainly bounds $M^*(\mathcal{T})$ from below. Surprisingly, if $\mathcal{T}$ is a d-regular network and the packets are randomly distributed, $M^*(\mathcal{T})$ is very close to this lower bound with high probability:

Theorem 4: If $\mathcal{T}$ is a d-regular network and the packets are randomly distributed, then

$$M^*(\mathcal{T}) < M_{LP}(\mathcal{T}) + n$$

with probability approaching 1 as the number of packets $k \to \infty$.

We make two important observations. First, the length of the interval in which $M^*(\mathcal{T})$ is concentrated is independent of k. Hence, even though the number of packets k may be extremely large, $M^*(\mathcal{T})$ can be estimated accurately. Second, as k grows large, $M^*(\mathcal{T})$ is dominated by the *local* topology of $\mathcal{T}$. This is readily seen since the constraints defining $M_{LP}(\mathcal{T})$ correspond only to nodes' immediate neighborhoods. The importance of the local neighborhood was also seen in [24] where network coding capacity for certain random networks is shown to concentrate around the expected number of nearest neighbors of the source and the terminals.

D. Large (Divisible) Packets

We now return to general networks with arbitrarily distributed packets. However, we now consider the case where packets are "large" and can be divided into several smaller pieces (e.g., packets actually correspond to large files). To formalize this, assume that each packet can be partitioned into t chunks of equal size, and transmissions can consist of a single chunk (as opposed to an entire packet). In this case, we say the packets are t-divisible. To illustrate this point more clearly, we return to Example 2, this time considering 2-divisible packets.

Example 5 (2-Divisible Packets): Let $\mathcal{T}$ be the network of Example 2 and split each packet into two halves: $p_i \to (p_i^{(1)}, p_i^{(2)})$. Denote this new network $\mathcal{T}'$ with corresponding sets of packets:

$$P_i' = \{p_1^{(1)}, p_1^{(2)} p_2^{(1)}, p_2^{(2)}, p_3^{(1)}, p_3^{(2)}\} \backslash \{p_i^{(1)}, p_i^{(2)}\}.$$

Three chunk transmissions allow universal recovery as follows: Node 1 transmits $p_2^{(2)} \oplus p_3^{(2)}$. Node 2 transmits $p_1^{(1)} \oplus p_3^{(1)}$. Node 3 transmits $p_1^{(2)} \oplus p_2^{(1)}$. It is readily verified from (2) that 3 chunk-transmissions are required to permit universal recovery. Thus, $M^*(\mathcal{T}') = 3$. Hence, if we were allowed to split the packets of Example 2 into two halves, it would suffice to transmit 3 chunks. Normalizing the number of transmissions by the number of chunks per packet, we say that universal recovery can be achieved with 1.5 packet transmissions.

Motivated by this example, define $M_t^*(\mathcal{T})$ to be the minimum number of (normalized) packet-transmissions required to achieve universal recovery in the network $\mathcal{T}$ when packets are t-divisible. For the network $\mathcal{T}$ in Example 2, we saw above that $M_2^*(\mathcal{T}) = 1.5$.

It turns out, if packets are t-divisible and t is large, the cut-set bounds (2) are "nearly sufficient" for achieving universal recovery. To see this, let $M_{\text{cut-set}}(\mathcal{T})$ be the optimal value of the LP:

$$\text{minimize} \quad \sum_{i=1}^{n} x_i \tag{5}$$

$$\text{subject to:} \quad \sum_{i \in \partial(S)} x_i \geq \left| \bigcap_{i \in S} P_i^c \right| \quad \text{for each nonempty } S \subset V. \tag{6}$$

Clearly $M_{\text{cut-set}}(\mathcal{T}) \leq M_t^*(\mathcal{T})$ for any network $\mathcal{T}$ with t-divisible packets because the LP producing $M_{\text{cut-set}}(\mathcal{T})$ relaxes the integer constraints and is constrained only by (2) rather than the full set of constraints given in Theorem 1. However, there exist transmission schedules which can approach this lower bound. Stated more precisely:

Theorem 5: For any network $\mathcal{T}$, the minimum number of (normalized) packet-transmissions required to achieve universal recovery with t-divisible packets satisfies

$$\lim_{t\to\infty} M_t^*(\mathcal{T}) = M_{\text{cut-set}}(\mathcal{T}).$$

Precisely how large t is required to be in order to approach $M_{\text{cut-set}}(\mathcal{T})$ within a specified tolerance is not clear for general networks. However, an immediate consequence of Theorem 3 is that $t = n - 1$ is sufficient to achieve this lower bound with high probability when packets are randomly distributed in a fully connected network.

Finally, we remark that it is a simple exercise to construct examples where the cut-set bounds alone are not sufficient to characterize transmission schedules permitting universal recovery when packets are not divisible (e.g., a 4-node line network with packets p_1 and p_2 at the left-most and right-most nodes, respectively). Thus, t-divisibility of packets provides the additional degrees of freedom necessary to approach the cut-set bounds more closely.

E. Remarks

One interesting consequence of our results is that splitting packets does not significantly reduce the required number of packet-transmissions for many scenarios. Indeed, at most one transmission can be saved if the network is fully connected (under any distribution of packets). If the network is d-regular, we can expect to save fewer than n transmissions if packets are randomly distributed (in fact, at most one transmission per node). It seems possible that this result could be strengthened to include arbitrary distributions of packets in d-regular networks (as opposed to randomly distributed packets), but a proof has not been found.

The limited value of dividing packets has practical ramifications since there is usually some additional communication overhead associated with dividing packets (e.g. additional headers, etc. for each transmitted chunk are required). Thus, if the packets are very large, say each packet is a video file, our results imply that entire coded packets can be transmitted without significant loss, avoiding any additional overhead incurred by dividing packets.

IV. An Application: Secrecy Generation

In this section, we consider the setup of the cooperative data exchange problem for a fully connected network $\mathcal{T}$, but we consider a different goal. In particular, we wish to generate a secret-key among the nodes that cannot be derived by an eavesdropper privy to all of the transmissions among nodes. Also, like the nodes themselves, the eavesdropper is assumed to know the indices of the packets initially available to each node. The goal is to generate the maximum amount of "secrecy" that cannot be determined by the eavesdropper.

The theory behind secrecy generation among multiple terminals was originally established in [14] for a very general class of problems. Our results should be interpreted as a practical application of the theory originally developed in [14]. Indeed, our results and proofs are special cases of those in [14] which have been streamlined to deal with the scenario under consideration. The aim of the present section is to show how secrecy can be generated in a *practical* scenario. In particular, we show that it is possible to efficiently generate the maximum amount of secrecy (as established in [14]) among nodes in a fully connected network $\mathcal{T} = \{\mathcal{G}, P_1, \dots, P_n\}$. Moreover, we show that this is possible in the non-asymptotic regime (i.e., there are no ϵ's and we don't require the number of packets or nodes to grow arbitrarily large). Finally, we note that it is possible to generate perfect secrecy instead of ϵ-secrecy without any sacrifice.

A. Practical Secrecy Results

In this subsection, we state two results on secrecy generation. Proofs are again postponed until Section V. We begin with some definitions[2]. Let $\mathbf{F}$ denote the set of all transmissions (all of which are available to the eavesdropper by definition). A function K of the packets $\{p_1, \dots, p_k\}$ in the network is called a secret key (SK) if K is recoverable by all nodes after observing $\mathbf{F}$, and it satisfies the (perfect) secrecy condition

$$I(K; \mathbf{F}) = 0, \tag{7}$$

and the uniformity condition

$$\Pr\left(K = key\right) = \frac{1}{|\mathcal{K}|} \text{ for all } key \in \mathcal{K}, \tag{8}$$

[2] We attempt to follow the notation of [14] where appropriate.

where $\mathcal{K}$ is the alphabet of possible keys.

We define $C_{SK}(P_1, \ldots, P_n)$ to be the secret-key capacity for a particular distribution of packets. We will drop the notational dependence on $P_1, \ldots, P_n$ where it doesn't cause confusion. By this we mean that a secret-key K can be generated if and only if $\mathcal{K} = \mathbb{F}^{C_{SK}}$. In other words, the nodes can generate at most C_{SK} packets worth of secret-key. Our first result of this section is the following:

Theorem 6: The secret-key capacity is given by: $C_{SK}(P_1, \ldots, P_n) = k - M^*(\mathcal{T})$.

Next, consider the related problem where a subset $D \subset V$ of nodes is compromised. In this problem, the eavesdropper has access to $\mathbf{F}$ and P_i for $i \in D$. In this case, the secret-key should also be kept hidden from the nodes in D (or else the eavesdropper could also recover it). Thus, for a subset of nodes D, let $P_D = \bigcup_{i \in D} P_i$, and call K a private-key (PK) if it is a secret-key which is only recoverable by the nodes in $V \backslash D$, and also satisfies the stronger secrecy condition:

$$I(K; \mathbf{F}, P_D) = 0. \tag{9}$$

Similar to above, define $C_{PK}(P_1, \ldots, P_n, D)$ to be the private-key capacity for a particular distribution of packets and subset of nodes D. Again, we mean that a private-key K can be generated if and only if $\mathcal{K} = \mathbb{F}^{C_{PK}}$. In other words, the nodes in $V \backslash D$ can generate at most C_{PK} packets worth of private-key. Note that, since P_D is known to the eavesdropper, each node $i \in D$ can transmit its respective set of packets P_i without any loss of secrecy capacity.

Define a new network $\mathcal{T}_D = \{\mathcal{G}_D, \{P_i^{(D)}\}_{i \in V \backslash D}\}$ as follows. Let $\mathcal{G}_D$ be the complete graph on $V \backslash D$, and let $P_i^{(D)} = P_i \backslash P_D$ for each $i \in V \backslash D$. Thus, $\mathcal{T}_D$ is a fully connected network with $n - |D|$ nodes and $k - |P_D|$ packets. Our second result of this section is the following:

Theorem 7: The private-key capacity is given by:

$$C_{PK}(P_1, \ldots, P_n, D) = (k - |P_D|) - M^*(\mathcal{T}_D).$$

The basic idea for private-key generation is that the users in $V \backslash D$ should generate a secret-key from $\{p_1, \ldots, p_k\} \backslash P_D$.

By the definitions of the SK and PK capacities, Theorem 2 implies that it is possible to compute these capacities efficiently. Moreover, as we will see in the achievability proofs, these capacities can be achieved by performing coded cooperative data exchange amongst the nodes. Thus, the algorithm developed in Appendix A combined with the algorithm in [18] can be employed to efficiently solve the secrecy generation problem we consider.

We conclude this subsection with an example to illustrate the results.

Example 6: Consider again the network of Example 2 and assume $\mathbb{F} = \{0, 1\}$ (i.e., each packet is a single bit). The secret-key capacity for this network is 1 bit. After performing universal recovery, the eavesdropper knows p_2 and the parity $p_1 \oplus p_3$. A perfect secret-key is $K = p_1$ (we could alternatively use $K = p_3$). If any of the nodes are compromised by the eavesdropper, the private-key capacity is 0.

We remark that the secret-key in the above example can in fact be attained by all nodes using only one transmission (i.e., universal recovery is not a prerequisite for secret-key generation). However, it remains true that only one bit of secrecy can be generated.

V. Proofs of Main Results

A. Necessary and Sufficient Conditions for Universal Recovery

Proof of Theorem 1: This proof is accomplished by reducing the problem at hand to an instance of a single-source network coding problem and invoking the Max-Flow Min-Cut Theorem for network information flow [4].

First, fix the number of communication rounds r to be large enough to permit universal recovery. For a network $\mathcal{T}$, construct the network-coding graph $\mathcal{G}^{NC} = (V_{NC}, E_{NC})$ as follows. The vertex set, V_{NC} is defined as:

$$V_{NC} = \{s, u_1, \ldots, u_k\} \cup \bigcup_{j=0}^{r} \{v_1^j, \ldots, v_n^j\} \cup \bigcup_{j=1}^{r} \{w_1^j, \ldots, w_n^j\}.$$

The edge set, E_{NC}, consists of directed edges and is constructed as follows:

- For each $i \in [k]$, there is an edge of unit capacity[3] from s to u_i.
- If $p_i \in P_j$, then there is an edge of infinite capacity from u_i to v_j^0.
- For each $j \in [r]$ and each $i \in [n]$, there is an edge of infinite capacity from v_i^{j-1} to v_i^j.
- For each $j \in [r]$ and each $i \in [n]$, there is an edge of capacity b_i^j from v_i^{j-1} to w_i^j.
- For each $j \in [r]$ and each $i \in [n]$, there is an edge of infinite capacity from w_i^j to $v_{i'}^j$ iff $i' \in \Gamma(i)$.

The interpretation of this graph is as follows: the vertex u_i is introduced to represent packet p_i, the vertex v_i^j represents node i after the j^{th} round of communication, and the vertex w_i^j

[3] An edge of unit capacity can carry one field element $z \in \mathbb{F}$ per unit time.

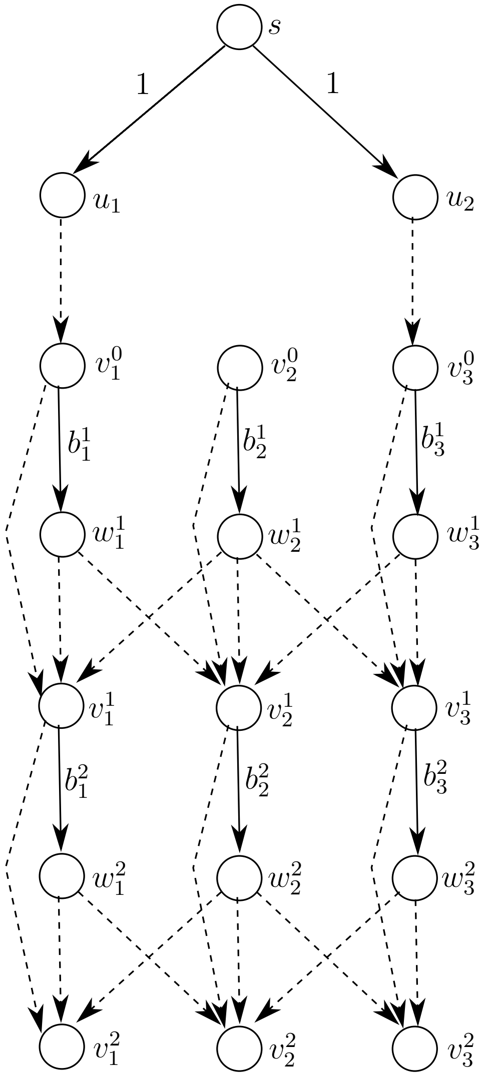

Fig. 4. The graph $\mathcal{G}^{NC}$ corresponding to the line network of Example 1. Edges represented by broken lines have infinite capacity. Edges with finite capacities are labeled with the corresponding capacity value.

represents the broadcast of node i during the j^{th} round of communication. If the b_i^j's are chosen such that the graph $\mathcal{G}^{NC}$ admits a network coding solution which supports a multicast of k units from s to $\{v_1^r, \ldots, v_n^r\}$, then this network coding solution also solves the universal recovery problem for the network $\mathcal{T}$ when node i is allowed to make at most b_i^j transmissions during the j^{th} round of communication. The graph $\mathcal{G}^{NC}$ corresponding to the line network of Example 1 is given in Figure 4.

We now formally prove the equivalence of the network coding problem on $\mathcal{G}^{NC}$ and the universal recovery problem defined by $\mathcal{T}$.

Suppose a set of encoding functions $\{f_i^j\}$ and a set decoding functions $\{\phi_i\}$ describe a transmission strategy which solves the universal recovery problem for a network $\mathcal{T}$ in r rounds of communication. Let b_i^j be the number of transmissions made by node i during the j^{th} round of communication, and let $\mathcal{I}_i^j$ be all the information known to node i prior to the j^{th} round of communication (e.g. $\mathcal{I}_i^1 = P_i$). The function f_i^j is the encoding function for user i during the j^{th} round of communication (i.e. $f_i^j(\mathcal{I}_i^j) \in \mathbb{F}^{b_i^j}$), and the decoding functions satisfy:

$$\phi_i\left(\mathcal{I}_i^r, \cup_{i' \in \Gamma(i)}\{f_{i'}^r(\mathcal{I}_{i'}^r)\}\right) = \{p_1, \dots, p_k\}.$$

Note that, given the encoding functions and the P_i's, the $\mathcal{I}_i^j$'s can be defined recursively as:

$$\mathcal{I}_i^{j+1} = \mathcal{I}_i^j \cup \bigcup_{i' \in \Gamma(i)} \{f_{i'}^j(\mathcal{I}_{i'}^j)\}.$$

The functions $\{f_i^j\}$ and $\{\phi_i\}$ can be used to generate a network coding solution which supports k units of flow from s to $\{v_1^r, \dots, v_n^r\}$ on $\mathcal{G}^{NC}$ as follows:

For each vertex $v \in V_{NC}$, let IN(v) be whatever v receives on its incoming edges. Let g_v be the encoding function at vertex v, and $g_v(e, \text{IN}(v))$ be the encoded message which vertex v sends along e (e is an outgoing edge from v).

If e is an edge of infinite capacity emanating from v, let $g_v(e, \text{IN}(v)) = \text{IN}(v)$.

Let s send p_i along edge (s, u_i). At this point, we have IN$(v_i^0) = P_i = \mathcal{I}_i^1$. For each $i \in [n]$, let $g_{v_i^0}((v_i^0, w_i^1), \text{IN}(v_i^0)) = f_i^1(\mathcal{I}_i^1)$. By a simple inductive argument, defining the encoding functions $g_{v_i^j}((v_i^j, w_i^{j+1}), \text{IN}(v_i^j))$ to be equal to f_i^{j+1} yields the result that IN$(v_i^r) = \left(\mathcal{I}_i^r, \cup_{i' \in \Gamma(i)}\{f_{i'}^r(\mathcal{I}_{i'}^r)\}\right)$. Hence, the decoding function ϕ_i can be used at v_i^r to allow error-free reconstruction of the k-unit flow.

The equivalence argument is completed by showing that a network coding solution which supports a k-unit multicast flow from s to $\{v_1^r, \dots, v_n^r\}$ on $\mathcal{G}^{NC}$ also solves the universal recovery problem on $\mathcal{T}$. This is argued in a similar manner as above, and is therefore omitted.

Since we have shown that the universal recovery problem on $\mathcal{T}$ is equivalent to a network coding problem on $\mathcal{G}^{NC}$, the celebrated max-flow min-cut result of Ahlswede et. al [4] is applicable. In particular, a fixed vector $\{b_i^j\}$ admits a solution to the universal recovery problem where node i makes at most b_i^j transmissions during the j^{th} round of communication if and only if any cut separating s from some v_i^r in $\mathcal{G}^{NC}$ has capacity at least k.

What remains to be shown is that the inequalities defining $\mathcal{R}_r(\mathcal{T})$ are satisfied if and only if any cut separating s from some v_i^r in $\mathcal{G}^{NC}$ has capacity at least k.

To this end, suppose we have a cut (S, S^c) satisfying $s \in S^c$ and $v_i^r \in S$ for some $i \in [n]$. We will modify the cut (S, S^c) to produce a new cut (S', S'^c) with capacity less than or equal to the capacity of the original cut (S, S^c).

Define the set $S_0 \subseteq [n]$ as follows: $i \in S_0$ iff $v_i^r \in S$ (by definition of S, we have that $S_0 \neq \emptyset$).

Initially, let $S' = S$. Modify the cut (S', S'^c) as follows:

M1) If $i \in \Gamma(S_0)$, then place w_i^r into S'.

M2) If $i \notin \Gamma(S_0)$, then place w_i^r into S'^c.

Modifications M1 and M2 are justified (respectively) by J1 and J2:

J1) If $i \in \Gamma(S_0)$, then there exists an edge of infinite capacity from w_i^r to some $v_{i'}^r \in S$. Thus, moving w_i^r to $\mathcal{S}'$ (if necessary) does not increase the capacity of the cut.

J2) If $i \notin \Gamma(S_0)$, then there are no edges from w_i^r to S, hence we can move w_i^r into S'^c (if necessary) without increasing the capacity of the cut.

Modifications M1 and M2 guarantee that $w_i^r \in S'$ iff $i \in \Gamma(S_0)$. Thus, assume that (S', S'^c) satisfies this condition and further modify the cut as follows:

M3) If $i \in S_0$, then place v_i^{r-1} into S'.

M4) If $i \notin \Gamma(S_0)$, then place v_i^{r-1} into S'^c.

Modifications M3 and M4 are justified (respectively) by J3 and J4:

J3) If $i \in S_0$, then there exists an edge of infinite capacity from v_i^{r-1} to $v_i^r \in S$. Thus, moving v_i^{r-1} to $\mathcal{S}'$ (if necessary) does not increase the capacity of the cut.

J4) If $i \notin \Gamma(S_0)$, then there are no edges from v_i^{r-1} to S' (since $w_i^r \notin S'$ by assumption), hence we can move v_i^{r-1} into S'^c (if necessary) without increasing the capacity of the cut.

At this point, define the set $S_1 \subseteq [n]$ as follows: $i \in S_1$ iff $v_i^{r-1} \in S'$. Note that the modifications of S' guarantee that S_1 satisfies $S_0 \subseteq S_1 \subseteq \Gamma(S_0)$.

This procedure can be repeated for each layer of the graph resulting in a sequence of sets $\emptyset \subsetneq S_0 \subseteq \cdots \subseteq S_r \subseteq [n]$ satisfying $S_j \subseteq \Gamma(S_{j-1})$ for each $j \in [r]$.

We now perform a final modification of the cut (S', S'^c):

M5) If $p_j \in \cup_{i \in S_r} P_i$, then place u_j into S'.

M6) If $p_j \notin \cup_{i \in S_r} P_i$, then place u_j into S'^c.

Modifications M5 and M6 are justified (respectively) by J5 and J6:

J5) If $p_j \in \cup_{i\in S_r} P_i$, then there is an edge of infinite capacity from u_j to S' and moving u_j into S' (if necessary) does not increase the capacity of the cut.

J6) If $p_j \notin \cup_{i\in S_r} P_i$, then there are no edges from u_j to S', hence moving u_j (if necessary) into S'^c cannot increase the capacity of the cut.

A quick calculation shows that the modified cut (S', S'^c) has capacity greater than or equal to k iff:

$$\sum_{j=1}^{r} \sum_{i\in S_j^c \cap \Gamma(S_{j-1})} b_i^{r+1-j} \geq \left| \bigcap_{i\in S_r} P_i^c \right|. \tag{10}$$

Since every modification of the cut either preserved or reduced the capacity of the cut, the original cut (S, S^c) also has capacity greater than or equal to k if the above inequality is satisfied. In Figure 5, we illustrate a cut (S, S^c) and its modified minimal cut (S', S'^c) for the graph $\mathcal{G}^{NC}$ corresponding to the line network of Example 1.

By the equivalence of the universal recovery problem on a network $\mathcal{T}$ to the network coding problem on $\mathcal{G}^{NC}$ and the max-flow min-cut theorem for network information flow, if a transmission scheme solves the universal recovery problem on $\mathcal{T}$, then the associated b_i^j's must satisfy the constraints of the form given by (10). Conversely, for any set of b_i^j's which satisfy the constraints of the form given by (10), there exists a transmission scheme using exactly those numbers of transmissions which solves the universal recovery problem for $\mathcal{T}$. Thus the constraints of (10), and hence the inequalities defining $\mathcal{R}_r(\mathcal{T})$, are satisfied if and only if any cut separating s from some v_i^r in $\mathcal{G}^{NC}$ has capacity at least k.

Remark 1: Since $\left|\bigcap_{i\in[n]} P_i^c\right| = 0$, constraints where $S_r = [n]$ are trivially satisfied. Therefore, we can restrict our attention to sequences of sets where $S_r \subsetneq [n]$.

■

B. Fully Connected Networks

Proof of Theorem 2: In the case where $\mathcal{T}$ is a fully connected network, we have that $S_j^c \cap \Gamma(S_{j-1}) = S_j^c$ for any nonempty $S \subset V$. Therefore, the constraints defining $\mathcal{R}_r(\mathcal{T})$ become:

$$\sum_{j=1}^{r} \sum_{i\in S_j^c} b_i^{r+1-j} \geq \left| \bigcap_{i\in S_r} P_i^c \right|. \tag{11}$$

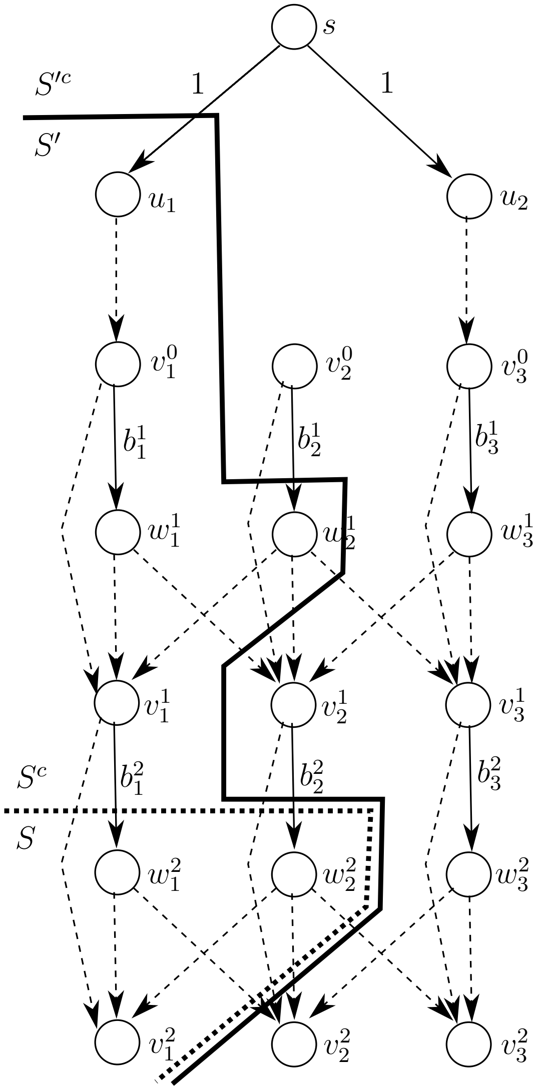

Fig. 5. The graph $\mathcal{G}^{NC}$ corresponding to the line network of Example 1 with original cut (S, S^c) and the corresponding modified minimal cut (S', S'^c). In this case, $S_0 = S_1 = S_2 = \{1\}$. Upon substitution into (10), this choice of S_0, S_1, S_2 yields the inequality $b_2^1 + b_2^2 \geq 1$.

Now, suppose a transmission schedule $\{b_i^j\} \in \mathcal{R}_r(\mathcal{T})$ and consider the modified transmission schedule $\{\tilde{b}_i^j\}$ defined by: $\tilde{b}_i^r = \sum_{j=1}^r b_i^j$ and $\tilde{b}_i^j = 0$ for $j < r$. By construction, $S_{j+1}^c \subseteq S_j^c$ in the constraints defining $\mathcal{R}_r(\mathcal{T})$. Therefore, using the definition of $\{\tilde{b}_i^j\}$, we have:

$$\sum_{i \in S_1^c} \tilde{b}_i^r \geq \sum_{j=1}^{r} \sum_{i \in S_j^c} b_i^{r+1-j} \geq \left| \bigcap_{i \in S_r} P_i^c \right| .$$

Thus the modified transmission schedule is also in $\mathcal{R}_r(\mathcal{T})$. Since $\left|\bigcap_{i \in S_1} P_i^c\right| \geq \left|\bigcap_{i \in S_r} P_i^c\right|$, when $\mathcal{T}$ is a fully connected network, it is sufficient to consider constraints of the form:

$$\sum_{i\in S^c} b_i^1 \geq \left|\bigcap_{i\in S} P_i^c\right| \quad \text{for all nonempty } S \subset V. \tag{12}$$

This proves the latter two statements of the theorem: that the cut-set constraints are necessary and sufficient for universal recovery when $\mathcal{T}$ is a fully connected network, and that a single round of communication is sufficient to achieve universal recovery with $M^*(\mathcal{T})$ transmissions.

With these results established, an optimal transmission schedule can be obtained by solving the following integer linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} b_i && (13)\\ \text{subject to:} \quad & \sum_{i\in S^c} b_i \geq \left|\bigcap_{i\in S} P_i^c\right| \quad \text{for each nonempty } S \subset V. \end{aligned}$$

In order to accomplish this, we identify $B_i \leftarrow P_i^c$ and set $w_i = 1$ for $i \in [n]$ and apply the submodular algorithm presented in Appendix A. ■

Now we consider fully connected networks in which packets are randomly distributed according to (1), which is parametrized by q. The proof of Theorem 3 requires the following lemma:

Lemma 1: If $0 < q < 1$ is fixed, then there exists some $\delta > 0$ such that the following inequality holds for all $\ell \in \{2, \ldots, n-1\}$:

$$\frac{n-\ell}{n-1} \geq \frac{(1-q)^\ell - (1-q)^n}{1-q-(1-q)^n} + \delta.$$

Proof: Applying Jensen's inequality to the strictly convex function $f(x) = (1-q)^x$ using the convex combination $\ell = \theta \cdot 1 + (1-\theta)\cdot n$ yields:

$$\frac{(1-q)^\ell - (1-q)^n}{1-q-(1-q)^n} < \frac{n-\ell}{n-1}.$$

Taking δ to be the minimum gap in the above inequality for the values $\ell \in \{2, \ldots, n-1\}$ completes the proof. ■

Proof of Theorem 3: We begin by showing that the LP

$$\text{minimize} \quad \sum_{i=1}^{n} b_i \tag{14}$$

$$\text{subject to:} \quad \sum_{i\in S^c} b_i \geq \left|\bigcap_{i\in S} P_i^c\right| \quad \text{for each nonempty } S \subset V. \tag{15}$$

has an optimal value of $\frac{1}{n-1}\sum_{i=1}^{n}|P_i^c|$ with high probability. To this end, note that the inequalities

$$\sum_{\substack{i=1\\ i\neq j}}^{n} b_i \geq |P_j^c| \quad \text{for } 1 \leq j \leq n. \tag{16}$$

are a subset of the inequality constraints (15). Summing both sides of (16) over $1 \leq j \leq n$ reveals that any feasible vector $b \in \mathbb{R}^n$ for LP (14)-(15) must satisfy:

$$\sum_{i=1}^{n} b_i \geq \frac{1}{n-1}\sum_{i=1}^{n}|P_i^c|. \tag{17}$$

This establishes a lower bound on the optimal value of the LP. We now identify a solution that is feasible with probability approaching 1 as $k \to \infty$ while achieving the lower bound of (17) with equality. To begin note that

$$\tilde{b}_j = \frac{1}{n-1}\sum_{i=1}^{n}|P_i^c| - |P_j^c| \tag{18}$$

is a solution to the system of linear equations given by (16) and achieves (17) with equality. Now, we prove that $(\tilde{b}_1, \ldots, \tilde{b}_n)$ is a feasible solution to LP (14) with high probability. To be specific, we must verify that

$$\sum_{i \in S^c} \tilde{b}_i \geq \left|\bigcap_{i \in S} P_i^c\right| \tag{19}$$

holds with high probability for all subsets $S \subset V$ satisfying $2 \leq |S| \leq n-1$ (the case $|S| = 1$ is satisfied by the definition of $\{\tilde{b}_i\}_{i=1}^n$). Substitution of (18) into (19) along with some algebra yields that the following equivalent conditions must hold:

$$\left(\frac{n-|S|}{n-1}\right)\sum_{i=1}^{n}\frac{1}{k}|P_i^c| - \sum_{i \in S^c}\frac{1}{k}|P_i^c| \geq \frac{1}{k}\left|\bigcap_{i \in S} P_i^c\right|. \tag{20}$$

To this end, note that for any S, $\left|\bigcap_{i\in S} P_i^c\right|$ is a random variable which can be expressed as $\left|\bigcap_{i\in S} P_i^c\right| = \sum_{j=1}^{k} X_j^S$, where X_j^S is an indicator random variable taking the value 1 if $p_j \in \bigcap_{i\in S} P_i^c$ and 0 otherwise. From (1) we have:

$$\Pr\left(X_j^S = 1\right) = \frac{(1-q)^{|S|} - (1-q)^n}{1-(1-q)^n}.$$

By the weak law of large numbers, for any $\eta > 0$:

$$\Pr\left(\left|\frac{1}{k}\left|\bigcap_{i\in S} P_i^c\right| - \frac{(1-q)^{|S|} - (1-q)^n}{1-(1-q)^n}\right| > \eta\right) < \epsilon_k, \tag{21}$$

where $\epsilon_k \to 0$ as $k \to \infty$. Thus, by the union bound, Lemma 1, and taking η sufficiently small, the following string of inequalities holds with arbitrarily high probability as $k \to \infty$:

$$\begin{aligned}
&\left(\frac{n-|S|}{n-1}\right)\sum_{i=1}^{n}\frac{1}{k}\left|P_i^c\right| - \sum_{i\in S^c}\frac{1}{k}\left|P_i^c\right| \\
&\geq \left(\frac{n-|S|}{n-1}\right)\left(\frac{(1-q)-(1-q)^n}{1-(1-q)^n} - (2n-1)\eta\right) \\
&\geq \frac{(1-q)^{|S|}-(1-q)^n}{1-(1-q)^n} + \eta \\
&\geq \frac{1}{k}\left|\bigcap_{i\in S} P_i^c\right|.
\end{aligned}$$

These steps are justified as follows: for η sufficiently small the first and last inequalities hold with high probability by (21), and the second inequality follows from Lemma 1 with $\ell = |S|$. This proves that (20) holds, and therefore $(\tilde{b}_1, \ldots, \tilde{b}_n)$ is a feasible solution to LP (14) with high probability. Now, taking Corollary 1 in Appendix A together with Theorem 2 completes the proof. ∎

C. d-Regular Networks

Lemma 2: Assume packets are randomly distributed in a d-regular network $\mathcal{T}$. For any $\epsilon > 0$, there exists an optimal solution x^* to LP (3-4) which satisfies

$$\left\| x^* - \frac{1}{d}\mathbb{E}[|P_1^c|]\mathbb{1} \right\|_\infty < \epsilon k$$

with probability approaching 1 as $k \to \infty$, where $\mathbb{E}$ indicates expectation.

Proof: Let $\vec{P} = (|P_1^c|, \ldots, |P_n^c|)^T$ and let A be the adjacency matrix of $\mathcal{G}$ (i.e., $a_{i,j} = 1$ if $(i,j) \in E$ and 0 otherwise). Observe that A is symmetric and $A\mathbb{1} = d\mathbb{1}$, where $\mathbb{1}$ denotes a column vector of 1's. With this notation, LP (3) can be rewritten as:

$$\begin{aligned}
\text{minimize} \quad & \mathbb{1}^T x \\
\text{subject to:} \quad & Ax \succeq \vec{P},
\end{aligned} \tag{22}$$

where "$a \succeq b$" for vectors $a, b \in \mathbb{R}^n$ means that $a_i \geq b_i$ for $i = 1, \ldots, n$.

Let A^+ denote the Moore-Penrose pseudoinverse of A. Observe that the linear least squares solution to $Ax \approx \vec{P}$ is given by:

$$\begin{aligned}\bar{x}_{LS} &= A^+\vec{P}\\ &= A^+\mathbb{E}\vec{P} \ + A^+\left(\vec{P}-\mathbb{E}\vec{P}\right)\\ &= \frac{1}{d}\mathbb{E}\vec{P} + A^+\left(\vec{P}-\mathbb{E}\vec{P}\right).\end{aligned}$$

For the last step above, note that $\mathbb{E}\vec{P}$ is an eigenvector of A with eigenvalue d so $\mathbb{E}\vec{P}$ will also be an eigenvector of A^+ with eigenvalue $\frac{1}{d}$. Hence,

$$\begin{aligned}\|x_{LS} - \frac{1}{d}\mathbb{E}\vec{P}\|_2 &= \|A^+\left(\vec{P}-\mathbb{E}\vec{P}\right)\|_2\\ &\le \|A^+\|_2\|\vec{P}-\mathbb{E}\vec{P}\|_2.\end{aligned}$$

Combining this with the triangle inequality implies that, for any vector y,

$$\begin{aligned}\|y - \frac{1}{d}\mathbb{E}\vec{P}\|_\infty &\le \|y-\bar{x}_{LS}\|_\infty + \|\bar{x}_{LS} - \frac{1}{d}\mathbb{E}\vec{P}\|_\infty\\ &\le \|y-\bar{x}_{LS}\|_\infty + \|\bar{x}_{LS} - \frac{1}{d}\mathbb{E}\vec{P}\|_2\\ &\le \|y-\bar{x}_{LS}\|_\infty + \|A^+\|_2\|\vec{P}-\mathbb{E}\vec{P}\|_2.\end{aligned}$$

Therefore, Lemma 7 (see Appendix B) guarantees the existence of an optimal solution x^* to LP (22) (and consequently LP (3)) which satisfies:

$$\begin{aligned}\|x^* - \frac{1}{d}\mathbb{E}\vec{P}\|_\infty &\le \|x^*-\bar{x}_{LS}\|_\infty + \|A^+\|_2\|\vec{P}-\mathbb{E}\vec{P}\|_2\\ &\le c_A\|A\bar{x}_{LS} - \vec{P}\|_2 + \|A^+\|_2\|\vec{P}-\mathbb{E}\vec{P}\|_2\\ &\le c_A\|\frac{1}{d}A\mathbb{E}\vec{P} - \vec{P}\|_2 + \|A^+\|_2\|\vec{P}-\mathbb{E}\vec{P}\|_2\\ &= c_A\|\mathbb{E}\vec{P} - \vec{P}\|_2 + \|A^+\|_2\|\vec{P}-\mathbb{E}\vec{P}\|_2,\end{aligned}$$

where c_A is a constant depending only on A. By the weak law of large numbers, $\|\vec{P}-\mathbb{E}\vec{P}\|_2 \le \epsilon k$ with probability tending to 1 as $k \to \infty$ for any $\epsilon > 0$. Noting that $\mathbb{E}\vec{P} = \mathbb{E}[|P_1^c|]\mathbb{1}$ completes the proof. ■

Proof of Theorem 4: We begin with some observations and definitions:

- First, recall that our model for randomly distributed packets (1) implies that

$$\mathbb{E}\left[\left|\bigcap_{i\in S} P_i^c\right|\right] = k\frac{(1-q)^{|S|}-(1-q)^n}{1-(1-q)^n} \quad \text{for all nonempty } S \subset V. \tag{23}$$

- With this in mind, there exists a constant $c_q > 0$ such that
$$\mathbb{E}\left[|P_1^c|\right] \geq (1+c_q)\mathbb{E}\left[\left|\bigcap_{i\in S} P_i^c\right|\right] \quad \text{for all } S \subset V, |S| \geq 2. \tag{24}$$
- Next, Lemma 1 implies the existence of a constant $\delta_q > 0$ such that for any $S \subset V$ with $2 \leq |S| \leq n-1$:
$$\frac{n-|S|}{n-1} \geq \frac{(1-q)^{|S|}-(1-q)^n}{(1-q)-(1-q)^n} + \delta_q = \frac{\mathbb{E}\left[|\bigcap_{i\in S} P_i^c|\right]}{\mathbb{E}\left[|P_1^c|\right]} + \delta_q. \tag{25}$$
- The weak law of large numbers implies that
$$\left(1+\frac{\min\{\delta_q, c_q\}}{4}\right)\mathbb{E}\left[\left|\bigcap_{i\in S} P_i^c\right|\right] \geq \left|\bigcap_{i\in S} P_i^c\right| \tag{26}$$
with probability approaching 1 as $k \to \infty$.
- Finally, for the proof below, we will take the number of communication rounds sufficiently large to satisfy
$$r \geq \max\left\{\frac{2d}{n\delta_q}, \frac{2n(1+c_q)}{dc_q}\right\}. \tag{27}$$

Fix $\epsilon > 0$. Lemma 2 guarantees that there exists an optimal solution x^* to LP (3) satisfying
$$\left\| x^* - \frac{1}{d}\mathbb{E}[|P_1^c|]\mathbb{1}\right\|_\infty < \epsilon k \tag{28}$$
with probability tending to 1 in k. Now, it is always possible to construct a transmission schedule $\{b_i^j\}$ which satisfies $\sum_j b_i^j = \lceil x_i^* \rceil$ and $\lfloor \frac{1}{r} x_i^* \rfloor \leq b_i^j \leq \lceil \frac{1}{r} x_i^* \rceil$ for each i, j. Observe that $\sum_{i,j} b_i^j < n + \sum_i x_i^*$. Thus, proving that $\{b_i^j\} \in \mathcal{R}_r(\mathcal{T})$ with high probability will prove the theorem.

Since the network is d-regular, $|\partial(S)| \geq d$ whenever $|S| \leq n-d$ and $|\partial(S_1)| \geq n - |S_2|$ whenever $|S_2| \geq n-d$ and $S_1 \subseteq S_2$. We consider the cases where $2 \leq |S_r| \leq n-d$ and $n-d < |S_r| \leq n-1$ separately. The case where $|S_r| = 1$ coincides precisely with the constraints (4), and hence is satisfied by definition of $\{b_i^j\}$.

Considering the case where $2 \le |S_r| \le n-d$, we have the following string of inequalities:

$$\begin{aligned}
\sum_{j=1}^{r} \sum_{i \in S_j^c \cap \Gamma(S_{j-1})} b_i^{(r+1-j)} &\ge \sum_{j=1}^{r} \sum_{i \in S_j^c \cap \Gamma(S_{j-1})} \left\lfloor \frac{1}{r} x_i^* \right\rfloor && (29) \\
&\ge \frac{1}{r} \sum_{j=1}^{r} \sum_{i \in S_j^c \cap \Gamma(S_{j-1})} x_i^* - nr && (30) \\
&= \frac{1}{r} \sum_{j=1}^{r} \sum_{i \in \partial(S_{j-1})} x_i^* - \frac{1}{r} \sum_{i \in S_r \cap S_0^c} x_i^* - nr && (31) \\
&\ge \frac{1}{r} \sum_{j=1}^{r} \sum_{i \in \partial(S_{j-1})} \frac{1}{d} \mathbb{E}[|P_1^c|] - \frac{1}{r} \sum_{i \in S_r \cap S_0^c} \frac{1}{d} \mathbb{E}[|P_1^c|] - nk\epsilon - nr && (32) \\
&\ge \frac{1}{rd} \mathbb{E}[|P_1^c|] \left(\sum_{j=1}^{r} |\partial(S_{j-1})| - n \right) - nr(k\epsilon + 1) && (33) \\
&\ge \frac{1+c_q}{rd} \mathbb{E}\left[\left| \bigcap_{i \in S_r} P_i^c \right| \right] \left(\sum_{j=1}^{r} |\partial(S_{j-1})| - n \right) - nr(k\epsilon + 1) && (34) \\
&\ge \frac{1+c_q}{rd} \mathbb{E}\left[\left| \bigcap_{i \in S_r} P_i^c \right| \right] (rd - n) - nr(k\epsilon + 1) && (35) \\
&\ge \left(1 + \frac{c_q}{2}\right) \mathbb{E}\left[\left| \bigcap_{i \in S_r} P_i^c \right| \right] - nr(k\epsilon + 1) && (36) \\
&\ge \left(1 + \frac{c_q}{4}\right) \mathbb{E}\left[\left| \bigcap_{i \in S_r} P_i^c \right| \right] && (37) \\
&\ge \left| \bigcap_{i \in S_r} P_i^c \right|. && (38)
\end{aligned}$$

The above string of inequalities holds with probability tending to 1 as $k \to \infty$. They can be justified as follows:

- (29) follows by definition of $\{b_i^j\}$.
- (30) follows since $\lfloor \frac{1}{r} x_i^* \rfloor \ge \frac{1}{r} x_i^* - 1$ and $|S_j^c \cap \Gamma(S_{j-1})| \le n$.
- (31) follows from writing $\cup_{j=1}^r S_j^c \cap \Gamma(S_{j-1})$ as $\left(\cup_{j=1}^r \partial(S_{j-1})\right) \setminus (S_r \cap S_0^c)$ and expanding the sum.
- (32) follows from (28).
- (33) is true since $|S_0^c \cap S_r| \le n$.
- (34) follows from (24).

- (35) follows from $|\partial(S_{j-1})| \geq d$ by d regularity and the assumption that $2 \leq |S_r| \leq n-d$.
- (36) follows from our choice of r given in (27).
- (37) follows since $\frac{c_q}{4}\mathbb{E}\left[\left|\bigcap_{i\in S_r} P_i^c\right|\right] \geq nr(k\epsilon+1)$ with high probability for ϵ sufficiently small.
- (38) follows from (26).

Next, consider the case where $n-d \leq |S_r| \leq n-1$. Starting from (33), we obtain:

$$\begin{aligned}
\sum_{j=1}^{r}\sum_{i\in S_j^c\cap\Gamma(S_{j-1})} b_i^{(r+1-j)} &\geq \frac{1}{rd}\mathbb{E}[|P_1^c|]\left(\sum_{j=1}^{r}|\partial(S_{j-1})| - n\right) - nr(k\epsilon+1) && (39)\\
&\geq \frac{1}{rd}\mathbb{E}[|P_1^c|]\Big(r(n-|S_r|) - n\Big) - nr(k\epsilon+1) && (40)\\
&= \mathbb{E}[|P_1^c|]\left(\frac{n-|S_r|}{d} - \frac{n}{rd}\right) - nr(k\epsilon+1) && (41)\\
&\geq \mathbb{E}[|P_1^c|]\left(\frac{n-|S_r|}{n-1} - \frac{n}{rd}\right) - nr(k\epsilon+1) && (42)\\
&\geq \mathbb{E}[|P_1^c|]\left(\frac{\mathbb{E}\left[\left|\bigcap_{i\in S_r} P_i^c\right|\right]}{\mathbb{E}[|P_1^c|]} + \delta_q - \frac{n}{rd}\right) - nr(k\epsilon+1) && (43)\\
&\geq \mathbb{E}[|P_1^c|]\left(\frac{\mathbb{E}\left[\left|\bigcap_{i\in S_r} P_i^c\right|\right]}{\mathbb{E}[|P_1^c|]} + \frac{\delta_q}{2}\right) - nr(k\epsilon+1) && (44)\\
&\geq \left(1+\frac{\delta_q}{4}\right)\mathbb{E}\left[\left|\bigcap_{i\in S_r} P_i^c\right|\right] && (45)\\
&\geq \left|\bigcap_{i\in S_r} P_i^c\right|. && (46)
\end{aligned}$$

The above string of inequalities holds with probability tending to 1 as $k\to\infty$. They can be justified as follows:

- (39) is simply (33) repeated for convenience.
- (40) follows since $n-d \leq |S_r| \leq n-1$ and hence d-regularity implies that $|\partial(S_{j-1})| \geq (n-|S_r|)$.
- (42) follows since $d \leq n-1$.
- (43) follows from (25).
- (44) follows from from our definition of r given in (27).
- (45) follows since $\frac{\delta_q}{4}\mathbb{E}[P_1^c] \geq nr(k\epsilon+1)$ with high probability for ϵ sufficiently small.
- (46) follows from (26).

Thus, we conclude that, for ϵ sufficiently small, the transmission schedule $\{b_i^j\}$ satisfies each of the inequalities defining $\mathcal{R}_r(\mathcal{T})$ with probability tending to 1. Since the number of such inequalities is finite, an application of the union bound completes the proof that $\{b_i^j\} \in \mathcal{R}_r(\mathcal{T})$ with probability tending to 1 as $k \to \infty$. ■

D. Divisible Packets

Proof of Theorem 5: Fix any $\epsilon > 0$ and let x^* be an optimal solution to LP (5). Put $b_i = x_i^* + \epsilon$. Note that b_i is nonnegative. This follows by considering the set $S\backslash\{i\}$ in the inequality constraint (6), which implies $x_i^* \geq 0$.

Now, take an integer $r \geq \epsilon^{-1} n \max_{1\leq i\leq n} b_i$. If packets are t-divisible, we can find a transmission schedule $\{b_i^j\}$ such that $\frac{1}{r} b_i \leq b_i^j \leq \frac{1}{r} b_i + \frac{1}{t}$ for all $i \in [n], j \in [r]$.

Thus, for any $(S_0, \cdots, S_r) \in \mathcal{S}^{(r)}(\mathcal{G})$ we have the following string of inequalities:

$$\begin{aligned}
\sum_{j=1}^{r} \sum_{i\in S_j^c \cap \Gamma(S_{j-1})} b_i^{(r+1-j)} &\geq \frac{1}{r} \sum_{j=1}^{r} \sum_{i\in S_j^c \cap \Gamma(S_{j-1})} b_i \\
&= \frac{1}{r} \sum_{j=1}^{r} \sum_{i\in \partial(S_{j-1})} b_i - \frac{1}{r} \sum_{i \in S_0^c \cap S_r} b_i \\
&= \frac{1}{r} \sum_{j=1}^{r} \sum_{i\in \partial(S_{j-1})} x_i^* + \frac{\epsilon}{r} \sum_{j=1}^{r} |\partial(S_{j-1})| - \frac{1}{r} \sum_{i \in S_0^c \cap S_r} b_i \\
&\geq \frac{1}{r} \sum_{j=1}^{r} \left| \bigcap_{i \in S_{j-1}} P_j^c \right| + \epsilon - \frac{n}{r} \max_{1\leq i \leq n} b_i \\
&\geq \left| \bigcap_{i\in S_r} P_j^c \right|.
\end{aligned}$$

Hence, Theorem 1 implies that the transmission schedule $\{b_i^j\}$ is sufficient to achieve universal recovery. Noting that

$$\sum_{i,j} b_i^j \leq \sum_{i=1}^{n} b_i + \frac{nr}{t} \leq \sum_{i=1}^{n} x_i^* + n\left(\frac{r}{t} + \epsilon\right)$$

completes the proof of the theorem. ■

E. Secrecy Generation

In this subsection, we prove Theorems 6 and 7. We again remark that our proofs can be seen as special cases of those in [14] which have been adapted for the problem at hand. For notational convenience, define $P = \{p_1, \dots, p_k\}$. We will require the following lemma.

Lemma 3: Given a packet distribution $P_1, \dots, P_n$, let K be a secret-key achievable with communication $\mathbf{F}$. Then the following holds:

$$H(K|\mathbf{F}) = H(P) - \sum_{i=1}^{n} x_i. \tag{47}$$

for some vector $x = (x_1, \dots, x_n)$ which is feasible for the following ILP:

$$\text{minimize} \quad \sum_{i=1}^{n} x_i \tag{48}$$

$$\text{subject to:} \quad \sum_{i \in S} x_i \geq \left| \bigcap_{i \in S^c} P_i^c \right| \quad \text{for each nonempty } S \subset V. \tag{49}$$

Moreover, if K is a PK (with respect to a set D) and each node $i \in D$ transmits its respective set of packets P_i, then

$$H(K|\mathbf{F}) = H(P|P_D) - \sum_{i \in V \backslash D} x_i. \tag{50}$$

for some vector $x = (x_1, \dots, x_n)$ which is feasible for the ILP:

$$\text{minimize} \quad \sum_{i \in V \backslash D} x_i \tag{51}$$

$$\text{subject to:} \quad \sum_{i \in S} x_i \geq \left| \bigcap_{i \in S^c} P_i^c \right| \quad \text{for each nonempty } S \subset V \backslash D. \tag{52}$$

Remark 2: We remark that (49) and (52) are necessary and sufficient conditions for achieving universal recovery in the networks $\mathcal{T}$ and $\mathcal{T}_D$ considered in Theorems 6 and 7, respectively. Thus, the optimal values of ILPs (48) and (51) are equal to $M^*(\mathcal{T})$ and $M^*(\mathcal{T}_D)$, respectively.

Proof: We assume throughout that all entropies are with respect to the base-$|\mathbb{F}|$ logarithm (i.e., information is measured in packets). For this and the following proofs, let $\mathbf{F} = (F_1, \dots, F_n)$ and $F_{[1,i]} = (F_1, \dots, F_i)$, where F_i denotes the transmissions made by node i. For simplicity, our proof does not take into account interactive communication, but can be modified to do so. Allowing interactive communication does not change the results. See [14] for details.

Since K and $\mathbf{F}$ are functions of P:

$$H(P) = H(\mathbf{F}, K, P_1, \ldots, P_n) \tag{53}$$

$$= \sum_{i=1}^{n} H(F_i|F_{[1,i-1]}) + H(K|\mathbf{F}) + \sum_{i=1}^{n} H(P_i|\mathbf{F}, K, P_{[1,i-1]}). \tag{54}$$

Set $x_i = H(F_i|F_{[1,i-1]}) + H(P_i|\mathbf{F}, K, P_{[1,i-1]})$. Then, the substituting x_i into the above equation yields:

$$H(K|\mathbf{F}) = H(P) - \sum_{i=1}^{n} x_i. \tag{55}$$

To show that $x = (x_1, \ldots, x_n)$ is a feasible vector for ILP (48), we write:

$$\left|\bigcap_{i\in S^c} P_i^c\right| = H(P_S|P_{S^c}) \tag{56}$$

$$= H(\mathbf{F}, K, P_S|P_{S^c}) \tag{57}$$

$$= \sum_{i=1}^{n} H(F_i|F_{[1,i-1]}, P_{S^c}) + H(K|\mathbf{F}, P_{S^c}) + \sum_{i\in S} H(P_i|\mathbf{F}, K, P_{[1,i-1]}, P_{S^c\cap[i+1,n]}) \tag{58}$$

$$\leq \sum_{i\in S} H(F_i|F_{[1,i-1]}) + \sum_{i\in S} H(P_i|\mathbf{F}, K, P_{[1,i-1]}) \tag{59}$$

$$= \sum_{i\in S} x_i. \tag{60}$$

In the above inequality, we used the fact that conditioning reduces entropy, the fact that K is a function of $(\mathbf{F}, P_{S^c})$ for any $S \neq V$, and the fact that F_i is a function of P_i (by the assumption that communication is not interactive).

To prove the second part of the lemma, we can assume $D = \{1, \ldots, \ell\}$. The assumption that each node i in D transmits all of the packets in P_i implies $F_i = P_i$. Thus, for $i \in D$ we have $x_i = H(P_i|P_{[1,i-1]})$. Repeating the above argument, we obtain

$$H(K|\mathbf{F}) = H(P) - H(P_D) - \sum_{i\in V\setminus D} x_i \tag{61}$$

$$= H(P|P_D) - \sum_{i\in V\setminus D} x_i, \tag{62}$$

completing the proof of the lemma. ■

Proof of Theorem 6: Converse Part. Suppose K is a secret-key achievable with communication $\mathbf{F}$. Then, by definition of a SK and Lemma 3 we have

$$C_{SK} = H(K) = H(K|\mathbf{F}) = H(P) - \sum_{i=1}^{n} x_i \leq H(P) - M^*(\mathcal{T}) = k - M^*(\mathcal{T}). \tag{63}$$

Achievability Part. By definition, universal recovery can be achieved with $M^*(\mathcal{T})$ transmissions. Moreover, the communication $\mathbf{F}$ can be generated as a linear function of P (see the proof of Theorem 1 and [18]). Denote this linear transformation by $\mathbf{F} = \mathcal{L}P$. Note that $\mathcal{L}$ only depends on the indices of the packets available to each node, not the values of the packets themselves (see [18]). Let $\mathcal{P}_{\mathbf{F}} = \{P' : \mathcal{L}P' = \mathbf{F}\}$ be the set of all packet distributions which generate $\mathbf{F}$.

By our assumption that the packets are i.i.d. uniform from $\mathbb{F}$, each $P' \in \mathcal{P}_{\mathbf{F}}$ is equally likely given $\mathbf{F}$ was observed. Since $\mathbf{F}$ has dimension $M^*(\mathcal{T})$, $|\mathcal{P}_{\mathbf{F}}| = \mathbb{F}^{k-M^*(\mathcal{T})}$. Thus, we can set $\mathcal{K} = \mathbb{F}^{k-M^*(\mathcal{T})}$ and label each $P' \in \mathcal{P}_{\mathbf{F}}$ with a unique element in $\mathcal{K}$. The label for the actual P (which is reconstructed by all nodes after observing $\mathbf{F}$) is the secret-key. Thus, $C_{SK} \geq k - M^*(\mathcal{T})$.

We remark that this labeling can be done efficiently by an appropriate linear transformation mapping P to K. ■

Proof of Theorem 7: Converse Part. Suppose K is a private-key. Then, by definition of a PK and Lemma 3,

$$\begin{aligned} C_{PK} = H(K) = H(K|\mathbf{F}) = H(P|P_D) - \sum_{i \in V \setminus D} x_i \\ \leq H(P|P_D) - M^*(\mathcal{T}_D) = (k - |P_D|) - M^*(\mathcal{T}_D). \end{aligned}$$

Achievability Part. Let each node $i \in D$ transmit P_i so that we can update $P_j \leftarrow P_j \cup P_D$ for each $j \in V \setminus D$. Now, consider the universal recovery problem for only the nodes in $V \setminus D$. $M^*(\mathcal{T}_D)$ is the minimum number of transmissions required among the nodes in $V \setminus D$ so that each node in $V \setminus D$ recovers P. At this point, the achievability proof proceeds identically to the SK case. ■

VI. Concluding Remarks

In this paper, we derive necessary and sufficient conditions for achieving universal recovery in an arbitrarily connected network. For the case when the network is fully connected, we provide an efficient algorithm based on submodular optimization which efficiently solves the cooperative problem. This algorithm and its derivation yield tight concentration results for the case when packets are randomly distributed. Moreover, concentration results are provided when the network is d-regular and packets are distributed randomly. If packets are divisible, we prove that the traditional cut-set bounds are achievable. As a consequence of this and the concentration results, we show that splitting packets does not typically provide a significant benefit when the network is d-regular. Finally, we discuss an application to secrecy generation in the presence of an eavesdropper. We demonstrate that our submodular algorithm can be used to generate the maximum amount of secrecy in an efficient manner.

It is conceivable that the coded cooperative data exchange problem can be solved (or approximated) in polynomial time if the network is d-regular, but packets aren't necessarily randomly distributed. This is one possible direction for future work.

Acknowledgement

The authors would like to thank Kent Benson, Alex Sprintson, Pavan Datta, and Chi-Wen Su for the helpful conversations and suggestions which led to this paper.

Appendix A
An Efficiently Solvable Integer Linear Program

In this appendix, we introduce a special ILP and provide an efficient algorithm for solving it. This algorithm can be used to efficiently solve the cooperative data exchange problem when the underlying graph is fully-connected. We begin by introducing some notation[4].

Let $E = \{1, \dots, n\}$ be a finite set with n elements. We denote the family of all subsets of E by 2^E. We frequently use the compact notation $E \backslash U$ and $U + i$ to denote the sets $E \cap U^c$ and $U \cup \{i\}$ respectively. For a vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, define the corresponding functional $x : 2^E \to \mathbb{R}$ as:

$$x(U) := \sum_{i \in U} x_i, \text{ for } U \subseteq E. \tag{64}$$

[4] We attempt to keep the notation generic in order to emphasize that the results in this appendix are not restricted to the context of the cooperative data exchange problem.

Throughout this section, we let $\mathcal{F} = 2^E - \{\emptyset, E\}$ denote the family of nonempty proper subsets of E. Let $\mathcal{B} = \{B_1, \ldots, B_n\}$. No special structure is assumed for the B_i's except that they are finite.

With the above notation established, we consider the following Integer Linear Program (ILP) in this section:

$$\text{minimize} \left\{ \sum_{i \in E} w_i x_i : x(U) \geq \left| \bigcap_{i \in E \setminus U} B_i \right|, \forall\ U \in \mathcal{F}, x_i \in \mathbb{Z} \right\}. \tag{65}$$

It is clear that any algorithm that efficiently solves this ILP also solves ILP (13) by putting $B_i \leftarrow P_i^c$ and $w = \mathbb{1}$.

A. Submodular Optimization

Our algorithm for solving ILP (65) relies heavily on submodular function optimization. To this end, we give a very brief introduction to submodular functions here.

A function $g : 2^E \to \mathbb{R}$ is said to be submodular if, for all $X, Y \in 2^E$,

$$g(X) + g(Y) \geq g(X \cap Y) + g(X \cup Y). \tag{66}$$

Over the past three decades, submodular function optimization has received a significant amount of attention. Notably, several polynomial time algorithms have been developed for solving the Submodular Function Minimization (SFM) problem

$$\min \{g(U) : U \subseteq E\}. \tag{67}$$

We refer the reader to [25]–[27] for a comprehensive overview of SFM and known algorithms. As we will demonstrate, we can solve ILP (65) via an algorithm that iteratively calls a SFM routine. The most notable feature of SFM algorithms is their ability to solve problems with exponentially many constraints in polynomial time. One of the key drawbacks of SFM is that the problem formulation is very specific. Namely, SFM routines typically require the function g to be submodular on *all* subsets of the set E.

B. The Algorithm

We begin by developing an algorithm to solve an equality constrained version of ILP (65). We will remark on the general case at the conclusion of this section. To this end, let M be a positive integer and consider the following ILP:

$$\text{minimize} \quad w^T x \tag{68}$$

$$\text{subject to: } x(U) \geq \left| \bigcap_{i \in E \setminus U} B_i \right| \text{ for all } U \in \mathcal{F}, \text{ and} \tag{69}$$

$$x(E) = M. \tag{70}$$

Remark 3: We assume $w_i \geq 0$, else in the case without the equality constraint we could allow the corresponding $x_i \to +\infty$ and the problem is unbounded from below.

```
Algorithm A.1: SolveILP($\mathcal{B}, E, M, w$)

 comment: Define $f : 2^E \to \mathbb{R}$ as in equation (71).

 $x \leftarrow$ ComputePotentialX$(f, M, w)$
 if CheckFeasible$(f, x)$
   then return $(x)$
   else return (Problem Infeasible)
```

Theorem 8: Algorithm A.1 solves the equality constrained ILP (68) in polynomial time. If feasible, Algorithm A.1 returns an optimal x. If infeasible, Algorithm A.1 returns "Problem Infeasible".

Proof: The proof is accomplished in three steps:

1) First, we show that if our algorithm returns an x, it is feasible.
2) Second, we prove that if a returned x is feasible, it is also optimal.
3) Finally, we show that if our algorithm does not return an x, then the problem is infeasible.

Each step is given its own subsection. ■

Algorithm A.1 relies on three basic subroutines given below:

```
Algorithm A.2: ComputePotentialX($f, M, w$)

  comment: If feasible, returns $x$ satisfying (69) and (70) that minimizes $w^T x$.

  comment: Order elements of $E$ so that $w_1 \geq w_2 \geq \cdots \geq w_n$.

  for $i \leftarrow n$ to 2
    do { comment: Define $f_i(U) := f(U + i)$ for $U \subseteq \{i, \ldots, n\}$.
       { $x_i \leftarrow \mathrm{SFM}(f_i, \{i, \ldots, n\})$
  $x_1 \leftarrow M - \sum_{i=2}^{n} x_i$
  return ($x$)
```

```
Algorithm A.3: CheckFeasible($f, x$)

  comment: Check if $x(U) \leq f(U)$ for all $U \in \mathcal{F}$ with $1 \in U$.

  comment: Define $f_1(U) := f(U + 1)$ for $U \subseteq E$.

  if $\mathrm{SFM}(f_1, E) < 0$
    then return ( false )
    else return ( true )
```

```
Algorithm A.4: SFM($f, V$)

  comment: Minimize submodular function $f$ over groundset $V$. See [25] for details.

  $v \leftarrow \min\{f(U) : U \subseteq V\}$
  return ($v$)
```

C. Feasibility of a Returned x

In this section, we prove that if Algorithm A.1 returns a vector x, it must be feasible. We begin with some definitions.

Definition 6: A pair of sets $X, Y \subset E$ is called **crossing** if $X \cap Y \neq \emptyset$ and $X \cup Y \neq E$.

Definition 7: A function $g : 2^E \to \mathbb{R}$ is **crossing submodular** if

$$g(X) + g(Y) \geq g(X \cap Y) + g(X \cup Y)$$

for X, Y crossing.

We remark that minimization of crossing submodular functions is well established, however it involves a lengthy reduction to a standard submodular optimization problem. However, the crossing family $\mathcal{F}$ admits a straightforward algorithm, which is what we provide in Algorithm A.1. We refer the reader to [27] for complete details on the general case.

For M a positive integer, define

$$f(U) := M - \left|\bigcap_{i \in U} B_i\right| - x(U), \text{ for } U \in \mathcal{F}. \tag{71}$$

Lemma 4: The function f is crossing submodular on $\mathcal{F}$.

Proof: For $X, Y \in \mathcal{F}$ crossing:

$$\begin{aligned} f(X) + f(Y) &= M - \left|\bigcap_{i \in X} B_i\right| - x(X) + M - \left|\bigcap_{i \in Y} B_i\right| - x(Y) \\ &= M - \left|\bigcap_{i \in X} B_i\right| - x(X \cap Y) + M - \left|\bigcap_{i \in Y} B_i\right| - x(X \cup Y) \\ &\geq M - \left|\bigcap_{i \in X \cap Y} B_i\right| - x(X \cap Y) + M - \left|\bigcap_{i \in X \cup Y} B_i\right| - x(X \cup Y) \\ &= f(X \cap Y) + f(X \cup Y). \end{aligned}$$

■

Observe that, with f defined as above, the constraints of ILP (68) can be equivalently written as:

$$f(U) = M - \left|\bigcap_{i \in U} B_i\right| - x(U) \geq 0 \text{ for all } U \in \mathcal{F}, \text{ and} \tag{72}$$

$$x(E) = M. \tag{73}$$

Without loss of generality, assume the elements of E are ordered lexicographically so that $w_1 \geq w_2 \geq \cdots \geq w_n$. At iteration i in Algorithm A.2, $x_j = 0$ for all $j \leq i$. Thus, setting

$$x_i \leftarrow \min_{U \subseteq \{i, \ldots, n\}} \{f_i(U)\} \tag{74}$$

$$= \min_{U \subseteq \{i, \ldots, n\} : i \in U} \{f(U)\} \tag{75}$$

$$= \min_{U \subseteq \{i, \ldots, n\} : i \in U} \left\{ M - \left|\bigcap_{i \in U} B_i\right| - x(U) \right\} \tag{76}$$

and noting that the returned x satisfies $x(E) = M$, rearranging (76) guarantees that

$$x(E \backslash U) \geq \left| \bigcap_{i \in U} B_i \right|, \text{ for all } U \subseteq \{i, \ldots, n\}, i \in U \tag{77}$$

as desired. Iterating through $i \in \{2, \ldots, n\}$ guarantees (77) holds for $2 \leq i \leq n$.

Remark 4: In the feasibility check routine (Algorithm A.3), we must be able to evaluate $f_1(E)$. The reader can verify that putting $f(E) = 0$ preserves submodularity.

Now, in order for the feasibility check to return **true**, we must have

$$\begin{aligned} \min_{U \subseteq E} \{f_1(U)\} &= \min_{U \subseteq E : 1 \in U} \{f(U)\} && (78) \\ &= \min_{U \subseteq E : 1 \in U} \left\{ M - \left| \bigcap_{i \in U} B_i \right| - x(U) \right\} && (79) \\ &\geq 0, && (80) \end{aligned}$$

implying that

$$x(E \backslash U) \geq \left| \bigcap_{i \in U} B_i \right|, \text{ for all } U \subseteq E, 1 \in U. \tag{81}$$

Combining (77) and (81) and noting that $x(E) = M$ proves that x is indeed feasible. Moreover, x is integral as desired.

D. Optimality of a Returned x

In this section, we prove that if Algorithm A.1 returns a feasible x, then it is also optimal. First, we require two more definitions and a lemma.

Definition 8: A constraint of the form (72) corresponding to U is said to be **tight** for U if

$$f(U) = M - \left| \bigcap_{i \in U} B_i \right| - x(U) = 0. \tag{82}$$

Lemma 5: If x is feasible, X, Y are crossing, and their corresponding constraints are tight, then the constraints corresponding to $X \cap Y$ and $X \cup Y$ are also tight.

Proof: Since the constraints corresponding to X and Y are tight, we have

$$0 = f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y) \geq 0. \tag{83}$$

The first inequality is due to submodularity and the last inequality holds since x is feasible. This implies the result. ■

Definition 9: A family of sets $\mathcal{L}$ is **laminar** if $X, Y \in \mathcal{L}$ implies either $X \cap Y = \emptyset$, $X \subset Y$, or $Y \subset X$.

At iteration k ($1 < k \leq n$) of Algorithm A.2, let U_k be the set where (76) achieves its minimum. Note that $k \in U_k \subseteq \{k, \dots, n\}$. By construction, the constraint corresponding to U_k is tight. Also, the constraint $x(E) = M$ is tight. From the U_k's and E we can construct a laminar family as follows: if $U_j \cap U_k \neq \emptyset$ for $j < k$, then replace U_j with $\tilde{U}_j \leftarrow U_k \cup U_j$. By Lemma 5, the constraints corresponding to the sets in the newly constructed laminar family are tight. Call this family $\mathcal{L}$. For each $i \in E$, there is a unique smallest set in $\mathcal{L}$ containing i. Denote this set L_i. Since $k \in U_k \subseteq \{k, \dots, n\}$, $L_i \neq L_j$ for $i \neq j$. Note that $L_1 = E$ and $L_i \subset L_j$ only if $j < i$.

For each $L_i \in \mathcal{L}$ there is a unique smallest set L_j such that $L_i \subset L_j$. We call L_j the least upper bound on L_i.

Now, consider the dual linear program to (68):

$$\text{maximize} \quad -\sum_{U \in \mathcal{F}} \pi_U \left(M - \left| \bigcap_{i \in U} B_i \right| \right) - \pi_E M \tag{84}$$

$$\text{subject to:} \sum_{U \in \mathcal{F}: i \in U} \pi_U + \pi_E + w_i = 0, \text{ for } 1 \leq i \leq n \tag{85}$$

$$\pi_U \geq 0 \text{ for } U \in \mathcal{F}, \text{ and } \pi_E \text{ free.} \tag{86}$$

For each $L_i \in \mathcal{L}$, let the corresponding dual variable $\pi_{L_i} = w_j - w_i$, where L_j is the least upper bound on L_i. By construction, $\pi_{L_i} \geq 0$ since it was assumed that $w_1 \geq \cdots \geq w_n$. Finally, let $\pi_E = -w_1$ and $\pi_U = 0$ for $U \notin \mathcal{L}$.

Now, observe that:

$$\sum_{U \in \mathcal{F}: i \in U} \pi_U + \pi_E + w_i = 0 \tag{87}$$

as desired for each i. Thus, π is dual feasible. Finally, note that $\pi_U > 0$ only if $U \in \mathcal{L}$. However, the primal constraints corresponding to the sets in $\mathcal{L}$ are tight. Thus, (x, π) form a primal-dual feasible pair satisfying complementary slackness conditions, and are therefore optimal.

E. No Returned x = Infeasibility

Finally, we prove that if the feasibility check returns **false**, then ILP (68) is infeasible. Note by construction that the vector x passed to the feasibility check satisfies

$$M - \left|\bigcap_{i \in U} B_i\right| - x(U) \geq 0 \text{ for all nonempty } U \subseteq \{2, \ldots, n\}, \tag{88}$$

and $x(E) = M$. Again, let U_k be the set where (76) achieves its minimum and let $\mathcal{L}$ be the laminar family generated by these U_k's and E exactly as before. Again, the constraints corresponding to the sets in $\mathcal{L}$ are tight (this can be verified in a manner identical to the proof of Lemma 5). Now, since x failed the feasibilty check, there exists some exceptional set T with $1 \in T$ for which

$$M - \left|\bigcap_{i \in T} B_i\right| - x(T) < 0. \tag{89}$$

Generate a set L_T as follows: Initialize $L_T \leftarrow T$. For each $L_i \in \mathcal{L}, L_i \neq E$, if $L_T \cap L_i \neq \emptyset$, update $L_T \leftarrow L_T \cup L_i$. Now, we can add L_T to family $\mathcal{L}$ while preserving the laminar property. We pause to make two observations:

1) By an argument similar to the proof of Lemma 5, we have that
$$M - \left|\bigcap_{i \in L_T} B_i\right| - x(L_T) < 0.$$
2) The sets in $\mathcal{L}$ whose least upper bound is E form a partition of E. We note that L_T is a nonempty class of this partition. Call this partition $\mathcal{P}_{\mathcal{L}}$.

Again consider the dual constraints, however, let $w_i = 0$ (this does not affect feasibility). For each $L \in \mathcal{P}_{\mathcal{L}}$ define the associated dual variable $\pi_L = \alpha$, and let $\pi_E = -\alpha$. All other dual variables are set to zero. It is easy to check that this π is dual feasible. Now, the dual objective function becomes:

$$-\sum_{U \in \mathcal{F}} \pi_U \left((M - \left|\bigcap_{i \in U} B_i\right| \right) - \pi_E M = -\alpha \sum_{L \in \mathcal{P}_{\mathcal{L}}} \left(M - \left|\bigcap_{i \in L} B_i\right| - x(L) + x(L) \right) + \alpha M \tag{90}$$

$$= -\alpha \left(M - \left|\bigcap_{i \in L_T} B_i\right| - x(L_T) \right) - \alpha x(E) + \alpha M \tag{91}$$

$$= -\alpha \left(M - \left|\bigcap_{i \in L_T} B_i\right| - x(L_T) \right) \tag{92}$$

$$\rightarrow +\infty \text{ as } \alpha \rightarrow \infty. \tag{93}$$

Thus, the dual is unbounded and therefore the primal problem must be infeasible.

As an immediate corollary we obtain the following:

Corollary 1: The optimal values of the ILP:

$$\min\left\{x(E) : x(U) \geq \left|\cap_{i\in E\setminus U} B_i\right|, U \in \mathcal{F}, x_i \in \mathbb{Z}\right\}$$

and the corresponding LP relaxation:

$$\min\left\{x(E) : x(U) \geq \left|\cap_{i\in E\setminus U} B_i\right|, U \in \mathcal{F}, x_i \in \mathbb{R}\right\}$$

differ by less than 1.

Proof: Algorithm A.1 is guaranteed to return an optimal x if the intersection of the polytope and the hyperplane $x(E) = M$ is nonempty. Thus, if M^* is the minimum such M, then the optimal value of the LP must be greater than $M^* - 1$. ■

F. Solving the General ILP

Finally, we remark on how to solve the general case of the ILP without the equality constraint given in (65). First, we state a simple convexity result.

Lemma 6: Let $p^*_w(M)$ denote the optimal value of ILP (68) when the equality constraint is $x(E) = M$. We claim that $p^*_w(M)$ is a convex function of M.

Proof: Let M_1 and M_2 be integers and let $\theta \in [0,1]$ be such that $M_\theta = \theta M_1 + (1-\theta)M_2$ is an integer. Let $x^{(1)}$ and be $x^{(2)}$ optimal vectors that attain $p^*_w(M_1)$ and $p^*_w(M_2)$ respectively. Let $x^{(\theta)} = \theta x^{(1)} + (1-\theta)x^{(2)}$. By convexity, $x^{(\theta)}$ is feasible, though not necessarily integer. However, by the results from above, optimality is always attained by an integral vector. Thus, it follows that:

$$\theta p^*_w(M_1) + (1-\theta)p^*_w(M_2) = \theta w^T x^{(1)} + (1-\theta) w^T x^{(2)} = w^T x^{(\theta)} \geq p^*_w(M_\theta). \tag{94}$$

■

Noting that $p^*_w(M)$ is convex in M, we can perform bisection on M to solve the ILP in the general case. For our purposes, it suffices to have relatively loose upper and lower bounds on M since the complexity only grows logarithmically in the difference. A simple lower bound on M is given by $M \geq \max_i |B_i|$.

G. Complexity

Our aim in this paper is not to give a detailed complexity analysis of our algorithm. This is due to the fact that the complexity is dominated by the the SFM over the set E in Algorithm A.3. Therefore, the complexity of Algorithm A.1 is essentially the same as the complexity of the SFM solver employed.

However, we have performed a series of numerical experiments to demonstrate that Algorithm A.1 performs quite well in practice. In our implementation, we ran the Fujishige-Wolfe (FW) algorithm for SFM [28] based largely on a Matlab routine by A. Krause [29]. While the FW algorithm has not been proven to run in polynomial time, it has been shown to work quite well in practice [28] (similar to the Simplex algorithm for solving Linear Programs). Whether or not FW has worst-case polynomial complexity is an open problem to date. We remark that there are several SFM algorithms that run in strongly polynomial time which could be used if a particular application requires polynomially bounded worst-case complexity [25].

In our series of experiments, we chose $B_i \subset F$ randomly, where $|F| = 50$. We let $n = |E|$ range from 10 to 190 in increments of 10. For each value of n, we ran 10 experiments. The average computation time is shown in Figure 6, with error bars indicating one standard deviation. We consistently observed that the computations run in approximately $O(n^{1.85})$ time. Due to the iterative nature of the SFM algorithm, we anticipate that the computation time could be significantly reduced by implementing the algorithm in C/C++ instead of Matlab. However, the $O(n^{1.85})$ trend should remain the same. Regardless, we are able to solve the ILP problems under consideration with an astonishing 2^{190} constraints in approximately one minute.

APPENDIX B
A LINEAR PROGRAMMING APPROXIMATION LEMMA

Lemma 7: Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with nonnegative entries and all column sums equal to d. Let $\bar{x}_y$ be the vector of minimum Euclidean norm which minimizes $\|Ax_y - y\|_2$. There exists an optimal solution x^* to the linear program

$$\begin{aligned} \text{minimize} \quad & \mathbb{1}^T x \\ \text{subject to:} \quad & Ax \succeq y \end{aligned} \tag{95}$$

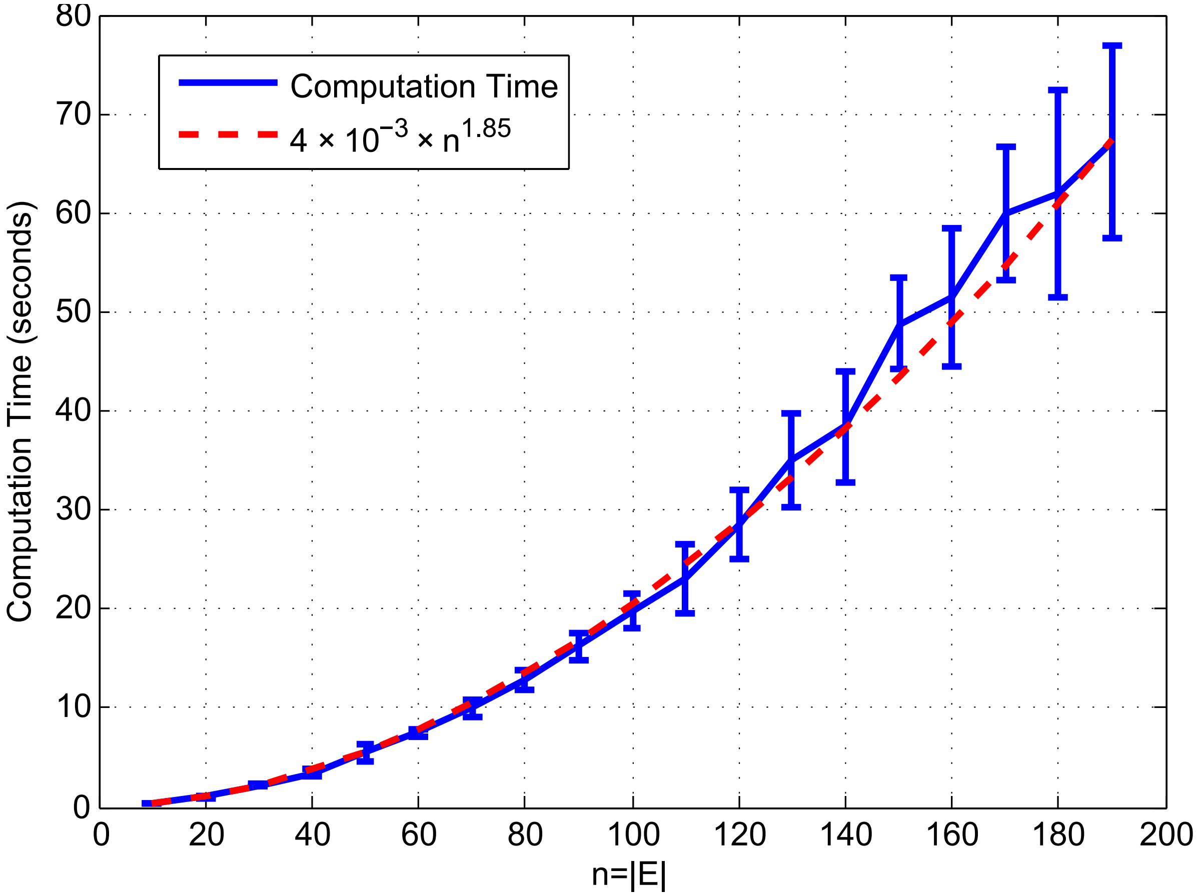

Fig. 6. Experimental results. For the red dotted line, the multiplicative constant α and exponent β were chosen to minimize the MSE $\sum_{i=1}^{n} |\log(\alpha n^{\beta}) - \log(\hat{m}_n)|^2$, where $\hat{m}_n$ is the sample mean of the computation times for $|E| = n$.

which satisfies

$$\|x^* - \bar{x}_y\|_\infty \leq c_A \|A\bar{x}_y - y\|_2,$$

where c_A is a constant depending only on A.

Proof of Lemma 7: To begin the proof, we make a few definitions. Let λ be the absolute value of the nonzero eigenvalue of A with smallest modulus (at least one exists since d is an eigenvalue). Define $\mathcal{N}(A)$ to be the nullspace of A, and let $\mathcal{N}^{\perp}(A)$ denote its orthogonal complement. Finally, let A^+ denote the Moore-Penrose pseudoinverse of A.

Fix $\bar{x}_y \in \mathbb{R}^n$, and note that x^* is an optimal solution to LP (95) if and only if $x^* - \bar{x}_y$ is an optimal solution to the linear program

$$\begin{aligned} \text{minimize} \quad & \mathbb{1}^T(x + \bar{x}_y) \\ \text{subject to:} \quad & A(x + \bar{x}_y) \succeq y \end{aligned}$$

with variable $x \in \mathbb{R}^n$. With this in mind, put $\bar{x}_y = A^+ y$ and define $b = y - A\bar{x}_y$. By definition of the pseudoinverse, $\bar{x}_y$ is the vector of minimum Euclidean norm which minimizes $\|Ax_y - y\|_2$. Moreover, $b \in \mathcal{N}(A)$.

Thus, in order to prove the lemma, it suffices to show the existence of an optimal solution x^* to the linear program

$$\begin{aligned} \text{minimize} \quad & \mathbb{1}^T x \\ \text{subject to:} \quad & Ax \succeq b \end{aligned} \tag{96}$$

which also satisfies the additional constraints

$$|x_i| \leq c_A \|b\|_2 \quad \text{for } i = 1, \ldots, n, \tag{97}$$

where c_A is a constant depending only on A.

Claim 1: There exists an optimal solution x^* to Linear Program (96) which satisfies

$$x_i^* \leq (d\lambda)^{-1} n \|b\|_\infty \quad \text{for } i = 1, \ldots, n. \tag{98}$$

The proof relies heavily on duality. The reader is directed to [30] or any other standard text for details.

To prove the claim, consider LP (96). By premultiplying the inequality constraint by $d^{-1}\mathbb{1}^T$ on both sides, we see that $\mathbb{1}^T x \geq d^{-1}\mathbb{1}^T b > -\infty$. Hence, the objective is bounded from below, which implies that strong duality holds. Thus, let $\tilde{z}$ be an optimal solution to the dual LP of (96):

$$\begin{aligned} \text{maximize} \quad & b^T z \\ \text{subject to:} \quad & Az = \mathbb{1} \\ & z \succeq 0 \end{aligned} \tag{99}$$

with dual variable $z \in \mathbb{R}^n$.

Next, consider the dual LP of (96) with the additional inequality constraints corresponding to (98):

$$\begin{aligned} \text{maximize} \quad & b^T z - (d\lambda)^{-1} n \|b\|_\infty \mathbb{1}^T y \\ \text{subject to:} \quad & Az = \mathbb{1} + y \\ & z \succeq 0 \\ & y \succeq 0 \end{aligned} \tag{100}$$

with dual variables $z \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$. Equivalently, by setting $z = \tilde{z} + \Delta z$ and observing that $y = A\Delta z$, we can write the dual LP (100) as

$$\begin{aligned} \text{maximize} \quad & b^T \tilde{z} + b^T \Delta z - (d\lambda)^{-1} n \|b\|_\infty \mathbb{1}^T A \Delta z \\ \text{subject to:} \quad & A\Delta z \succeq 0 \\ & \tilde{z} + \Delta z \succeq 0 \end{aligned} \tag{101}$$

with dual variables $\Delta z \in \mathbb{R}^n$. We prove the claim by showing that the dual LPs (99) and (101) have the same optimal value. Since strong duality holds, the corresponding primal problems must also have the same optimal value.

Without loss of generality, we can uniquely decompose $\Delta z = \Delta z_1 + \Delta z_2$ where $\Delta z_1 \in \mathcal{N}(A)$ and $\Delta z_2 \in \mathcal{N}^\perp(A)$. Since $b \in \mathcal{N}(A)$, we have $b^T \Delta z_2 = 0$ and we can rewrite (101) yet again as

$$\begin{aligned} \text{maximize} \quad & b^T \tilde{z} + b^T \Delta z_1 - (d\lambda)^{-1} n \|b\|_\infty \mathbb{1}^T A \Delta z_2 && (102) \\ \text{subject to:} \quad & A\Delta z_2 \succeq 0 \\ & \tilde{z} + \Delta z_1 + \Delta z_2 \succeq 0 && (103) \\ & \Delta z_1 \in \mathcal{N}(A), \Delta z_2 \in \mathcal{N}^\perp(A). \end{aligned}$$

By definition of λ, for any unit vector $u \in \mathcal{N}^\perp(A)$ with $\|u\|_2 = 1$ we have $\|Au\|_2 \geq \lambda$. Using this and the fact that $A\Delta z_2 \succeq 0$ for all feasible Δz_2, we have the following inequality:

$$\mathbb{1}^T A \Delta z_2 = \|A\Delta z_2\|_1 \geq \|A\Delta z_2\|_2 \geq \lambda \|\Delta z_2\|_2.$$

Thus, the objective (102) can be upper bounded as follows:

$$b^T \tilde{z} + b^T \Delta z_1 - (d\lambda)^{-1} n \|b\|_\infty \mathbb{1}^T A \Delta z_2 \leq b^T \tilde{z} + b^T \Delta z_1 - d^{-1} n \|b\|_\infty \|\Delta z_2\|_2. \tag{104}$$

Next, we obtain an upper bound on $b^T \Delta z_1$. To this end, observe that constraint (103) implies that $\tilde{z} + \Delta z_1 \succeq -\mathbb{1} \|\Delta z_2\|_\infty$. Motivated by this, consider the following ϵ-perturbed LP:

$$\begin{aligned} \text{minimize} \quad & -b^T v \\ \text{subject to:} \quad & \tilde{z} + v \succeq -\epsilon \mathbb{1} \\ & v \in \mathcal{N}(A). \end{aligned} \tag{105}$$

with variable v. Let $p^*(\epsilon)$ denote the optimal value of the ϵ-perturbed problem. First observe that $p^*(0) = 0$. To see this, note that if $\tilde{z} + v \succeq 0$, then $b^T v \leq 0$, else we would contradict the optimality of $\tilde{z}$ since $z = \tilde{z} + v$ is a feasible solution to the dual LP (99) in this case. Now, weak duality implies

$$-b^T v \geq p^*(\epsilon) \geq p^*(0) - \epsilon \mathbb{1}^T w^*, \tag{106}$$

where w^* corresponds to an optimal solution to the dual LP of the unperturbed primal LP (105), given by:

$$\begin{aligned} \text{maximize} \quad & -\tilde{z}^T (Aw - b) \\ \text{subject to:} \quad & Aw \succeq b. \end{aligned} \tag{107}$$

Hence, (106) implies that

$$b^T \Delta z_1 \leq \|\Delta z_2\|_\infty \mathbb{1}^T w^* \tag{108}$$

if $\Delta z_1, \Delta z_2$ are feasible for LP (102).

By definition of $\tilde{z}$, $\tilde{z}^T A = \mathbb{1}^T$, and hence a vector w^* is optimal for (107) if and only if it also optimizes:

$$\begin{aligned} \text{minimize} \quad & \mathbb{1}^T w \\ \text{subject to:} \quad & Aw \succeq b. \end{aligned}$$

Combining this with (108), we have

$$b^T \Delta z_1 \leq \|\Delta z_2\|_\infty \mathbb{1}^T w^* \leq \|\Delta z_2\|_\infty \mathbb{1}^T w$$

for any vector w satisfying $Aw \succeq b$. Trivially, $w = d^{-1} \|b\|_\infty \mathbb{1}$ satisfies this, and hence we obtain:

$$b^T \Delta z_1 \leq d^{-1} n \|b\|_\infty \|\Delta z_2\|_\infty.$$

Finally, we substitute this into (104) and see that

$$\begin{aligned} b^T z &\leq b^T \tilde{z} + d^{-1} n \|b\|_\infty \|\Delta z_2\|_\infty - d^{-1} n \|b\|_\infty \|\Delta z_2\|_2 \\ &\leq b^T \tilde{z} + d^{-1} n \|b\|_\infty \|\Delta z_2\|_2 - d^{-1} n \|b\|_\infty \|\Delta z_2\|_2 \\ &\leq b^T \tilde{z} \end{aligned}$$

for all vectors z which are feasible for the dual LP (100). This completes the proof of Claim 1.

Claim 2: There exists an optimal solution x^* to Linear Program (96) which satisfies

$$|x_i| \leq c_A \|b\|_2 \ \text{ for } i = 1, \dots, n \tag{109}$$

for some constant c_A depending only on A.

First note that $\|b\|_\infty \leq \|b\|_2$ for any $b \in \mathbb{R}^n$, hence it suffices to prove the claim for the infinity norm. Claim 1 shows that each of the x_i's can be upper bounded by $(d\lambda)^{-1} n \|b\|_\infty$ without affecting the optimal value of LP (96). To see the lower bound, let a_j^T be a row of A with entry $a_{ji} \geq d/n$ in the i^{th} coordinate (at least one exists for each i since the columns of A sum to d). Now, the inequality constraint $Ax \succeq b$ combined with the upper bound on each x_i implies:

$$a_{ji} x_i + (d - a_{ji}) \lambda^{-1} n \|b\|_\infty \geq a_j^T x \geq b_j \geq -\|b\|_\infty. \tag{110}$$

Since $a_{ji} \geq d/n$, (110) implies:

$$x_i \geq -\lambda^{-1} n (n-1) \|b\|_\infty.$$

Hence, we can take $c_A = \lambda^{-1} n \times \max\{n-1, d^{-1}\}$. This proves Claim 2, and, by our earlier remarks, proves the lemma. ■

References

[1] T. Courtade, B. Xie, and R. Wesel, "Optimal exchange of packets for universal recovery in broadcast networks," in *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, 31 2010-nov. 3 2010, pp. 2250 –2255.

[2] T. Courtade and R. Wesel, "Efficient universal recovery in broadcast networks," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 29 2010-oct. 1 2010, pp. 1542 –1549.

[3] ——, "Weighted universal recovery, practical secrecy, and an efficient algorithm for solving both," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, Oct. 2011.

[4] R. Ahlswede, N. Cai, S. yen Robert Li, and R. W. Yeung, "Network information flow," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 46, no. 4, pp. 1204–1216, 2000.

[5] S. yen Robert Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, 2003.

[6] S. El Rouayheb, M. Chaudhry, and A. Sprintson, “On the minimum number of transmissions in single-hop wireless coding networks,” in *Information Theory Workshop, 2007. ITW ’07. IEEE*, sept. 2007, pp. 120 –125.

[7] S. El Rouayheb, A. Sprintson, and P. Sadeghi, “On coding for cooperative data exchange,” in *Information Theory Workshop (ITW), 2010 IEEE*, Jan. 2010, pp. 1 –5.

[8] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, “A randomized algorithm and performance bounds for coded cooperative data exchange,” in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, june 2010, pp. 1888 –1892.

[9] ——, “Deterministic algorithm for coded cooperative data exchange,” in *ICST QShine*, Nov. 2010.

[10] D. Ozgul and A. Sprintson, “An algorithm for cooperative data exchange with cost criterion,” in *Information Theory and Applications Workshop (ITA), 2011*, feb. 2011, pp. 1 –4.

[11] Y. Birk and T. Kol, “Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients,” *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2825 – 2830, june 2006.

[12] E. Lubetzky and U. Stav, “Nonlinear index coding outperforming the linear optimum,” *Information Theory, IEEE Transactions on*, vol. 55, no. 8, pp. 3544 –3551, aug. 2009.

[13] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim, “Broadcasting with side information,” in *Foundations of Computer Science, 2008. FOCS ’08. IEEE 49th Annual IEEE Symposium on*, oct. 2008, pp. 823 –832.

[14] I. Csiszar and P. Narayan, “Secrecy capacities for multiple terminals,” *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 3047 – 3061, dec. 2004.

[15] C. Ye and P. Narayan, “Secret key and private key constructions for simple multiterminal source models,” in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, sept. 2005, pp. 2133 –2137.

[16] C. Ye and A. Reznik, “A simple secret key construction system for broadcasting model,” in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, march 2010, pp. 1 –6.

[17] R. M. Karp, “Reducibility Among Combinatorial Problems,” in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[18] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, “Polynomial time algorithms for multicast network code construction,” *Information Theory, IEEE Transactions on*, vol. 51, no. 6, pp. 1973 – 1982, june 2005.

[19] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413 –4430, oct. 2006.

[20] T. Halford, K. Chugg, and A. Polydoros, “Barrage relay networks: System amp; protocol design,” in *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, sept. 2010, pp. 1133 –1138.

[21] T. Halford and K. Chugg, “Barrage relay networks,” in *Information Theory and Applications Workshop (ITA), 2010*, 31 2010-feb. 5 2010, pp. 1 –8.

[22] T. Halford and G. Hwang, “Barrage relay networks for unmanned ground systems,” in *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, 31 2010-nov. 3 2010, pp. 1274 –1280.

[23] A. Blair, T. Brown, K. Chugg, T. Halford, and M. Johnson, “Barrage relay networks for cooperative transport in tactical manets,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, nov. 2008, pp. 1 –7.

[24] A. Ramamoorthy, J. Shi, and R. D. Wesel, “On the capacity of network coding for random networks,” *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 51, no. 8, pp. 2878–2885, 2005.

[25] S. McCormick, *Submodular Function Minimization. In Discrete Optimization, K. Aardal, G. Nemhauser, and R. Weismantel, eds. Handbooks in Operations Research and Management Science*. Elsevier, 2005, vol. 12.

[26] S. Fujishige, *Submodular Functions and Optimization*, 2nd ed. Berlin: Elsevier Science, 2010.

[27] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*. Berlin: Springer-Verlag, 2003.

[28] S. Fujishige, T. Hayashi, and S. Isotani, "The minimum-norm-point algorithm applied to submodular function minimization and," in *Kyoto University, Kyoto Japan*, 2006.

[29] A. Krause and S. Sonnenburg, "Sfo: A toolbox for submodular function optimization, the," *Journal of Machine Learning Research*, pp. 1141–1144, 2010.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.